Design for A Thinking Machine

Fifth Draft

Application of the Pi-Space Physics Theory

The Metropolis Draft

Brady Thinking Machine Design V0.5

Contents

Introduction

Inspiration for the Fifth Draft 'Metropolis'

In the 1920s Fritz Lang created a highly stylized futuristic city called 'Metropolis' which is a Utopian city on top with an underclass living beneath eking out a miserable existence. Even though the movie is silent with cards for spoken text, it has some stunning visuals and still holds up quite well conceptually in my opinion.

The Drafts

Note: This is the last Free-To-View Document Draft. All subsequent document drafts will be Pay-To-View.

Fifth Draft 'Metropolis' deals with the idea of predict-planning in a dynamic virtual-physical environment and examining Gaming Engines and the like to achieve intelligent decision making and movements. It also includes Functional Messaging and Home Brew and also Visual Functional Messaging.

Fourth Draft 'Copenhagen' included the use of Gaming Strategies to achieve Strategic and Tactical goals and a beginning to actual software prototypes.

Third Draft included the concept of Advanced Operators and Generic Algorithms, the LOS and the HOS interacting to form a being based on Equilibrium.

Second Draft included Appendices and the idea of Typed Actors and some prototype requirements.

First Draft were initial thoughts and mainly research on the medical aspects of the human body and brain and the virtual and the physical being.

Software Location

At present the tutorial examples are on github. This is a work in progress.

https://github.com/thinkingmachine/thinkingmachine

Abstract

This document asserts that the human brain is a mechanism to virtualize reality using an advanced messaging protocol and also contains a detailed virtual copy of the world that it is interacting with. This document highlights the job of the brain with the idea that it can be copied for a Cybernetic purpose. Presently, it is thought that Neural Nets are mainly the key to machine intelligence. This document asserts that the key to an intelligent robotic system is mainly in producing a technical system which visualizes reality into a "Window In The Mind" and here one can then use standard computer techniques for movement and collision detection using some kind of software based physics engine containing a virtual Self mirroring the physical self. A messaging algebra is also required for semantic processing of the virtual world and the body proper.

Therefore this simplifies the construction of so called Intelligent Machines where we are not so concerned about Neural Nets doing everything. The Virtual view of the Inner Window shall be obtained by using a Physics Engine. In this case, the Physics Engine chosen shall be Pi-Space Physics Engine which is based on the Pi-Space Physics Theory but it does not have to be.

This document also asserts that the Evolutionary aspect of DNA is part of the code of life but not its only intelligent aspect. This document asserts that life is essentially a multi-cell based organism containing the code for a virtual Self and a physical self which has a learning mechanism where improvements can be recoded into the generic program dynamically and across generations. Importantly, this document also asserts that the true key to intelligence in life forms is based on an Advanced Messaging algebra in combination with specialized Apps between the cells forming what we regard as higher intelligence.

The goal therefore of this document is to reverse engineer DNA into a general design pattern which is computationally efficient and can be used as the basis for life of an Android. The document will also discuss all the use major cases of life and flesh out the theoretical messaging algebra required not just for the brain but the body in general. The idea is not to figure out the exact messaging mechanism in the brain but have a theoretical solution which matches this behavior based on the Messaging Algebra and known use cases. It's the job of real neuro-scientists to figure out the exact messaging.

Also, the document asserts that DNA is a program that runs what is termed the HOS, or Human Operating System using an advanced Messaging Algebra between zones of specialized behavior. I realize that some folks may think means I am inferring we are

biological machines and may have some trouble accepting this concept. Be that as it may, the idea is to reverse engineer life into a machine format so it therefore makes sense to use machine concepts for the purpose of this document.

Note: The idea behind this creative endeavor is to **improve** peoples' lives with the aid of machines. The idea is **not** to figure out a technology to take their jobs from them. The social ramifications of any new technology needs to be measured up carefully by politicians and society in general. My own personal opinion is that we need at the very minimum to make our machines caring.

Part of the goal of this document is to reverse engineer the body based on some of its well known problems and features. These problems and features give us an insight into how the body is architected. This is also useful in my opinion for someone with an engineering background who does not know anything about human anatomy and is a non-medical introduction to aspects of the human body and brain. I am also learning this as I reverse engineer and any mistakes in understanding are of course my own.

This document is also a work in progress and may need clean up here and there. I am aware of this.

The longer term goal is not just to provide a theory but also to provide a software based solution as well which mirrors the messaging and app solution. Later then this would need to be applied to some kind of physical prototype. At present this is being worked on using software only.

Additionally, this theory can not only be applied to a humanoid structure but any structure that requires intelligence. This could be a flying device or a car or a building or a computer or a living space or some consumer item. This theory can therefore be used to add intelligence to any kind of object essentially where it makes sense to have it. Not every aspect of the messaging protocols will be needed and can be applied on a case by case need. The document is therefore a general specification and a list of simple builder apps are specified at the end of the document where one can build different pieces of the system and combine them where required.

Longer term, the plan is provide code for an Open Source Core Messaging system to which different Message Types can be attached. In some cases these Message Types can have hardware components and Dedicated Apps. However, life has evolved over millions of years to it make take several drafts before I seriously consider a complete software approach until I have covered the main topics in some detail. In draft 0.4 I have begun some simple prototypes.

The document also contains a set of exercises in the appendix for an interested student to try to design a software system themselves to achieve this design in term of pure product requirements. This is language independent and requirements based.

Additionally this design can also be used by Video Games which may be interested in designing intelligent Sims and the like.

Terms

HOS = Human Operating System

A.OS = Android Operating System

SC = System Consciousness (e.g. n bits/functional messages per second)

Sub-SC=Sub Consciousness (e.g. n million bits per second)

SC-GC = System Consciousness Graphics Card (it builds a 3D world view based on Visual Algebra)

SC-RR = System Conscious Reality Room (The room we live in when we are awake which is built by the SC-GC)

SC-PPR = System Conscious Predict Plan Room (Where we try out new ideas based on current RR)

HMIC = Human Machine Interface

HOSL = Human OS Language for Functional Messaging

OSL = Operating System Functional Messaging Language used by Android (Sounds like 'Ozzel')

App = A DNA Gene fragment that implements some task based on an Associated Functional Message. Sample tasks can be to co-ordinate other Apps switching them on or off to achieve some design goal or to build either a virtual or a physical system.

Message Data = A unique identifier related to one or more Apps

Typed Actor = A Cell of a particular type e.g. a Neuron

Runtime = The component elements of a live Typed Actor such as which Apps are presently turned on or off and other state based information pertinent to the Cell type.

Msg-App-Validate = An Atomic Unit of Interaction in the DNA

Msg-App Package = The Equivalent of a human Chromosome

Engine = An Enzyme (or Similar Mechanism) which processes the Msg-App-Validate cycle in Real Time

Kernel = The Core Micro services which support all the OS application e.g. LOS and HOS.

LOS = The base OS which sits on top of the Kernel

Operator=Something which operates on a Functional Message which is based on LOS Set Theory

Home Brew = An example Functional Messaging language

Functional Message Data = The data contained in a Functional Message

Definitions

What is life? : That which contains internal and or external functional messaging paths with different distribution models and co-operating strategies for evolving actionable functional message types and physical forms to achieve necessary goals.

An actionable functional message type invokes one or more Apps using Real Time Engines which may in turn call one or more Functional Messages or Apps forming Chains of Functional Messages which may be circular in nature.

A Functional Message is a dynamic message based on category theory and functional programming which contains both logic and data which a Thinking Machine uses to communicate higher level concepts like logic and reason.

What is intelligence?

Intelligence is defined by one or more goals which have associated static and/or dynamic games that use a replicate-refine design pattern that bids for games using predictive costings and later actual costings and also uses game functional message filters.

How is evolution described in this document?

Evolution of life is the point at which LOS self-replication reaches Equilibrium via functional messaging with self-specialization to form a new life e.g. HOS.

A dynamic game defines a game which can evolve based on existing and/or new functional message criteria.

Life communicates via Functional Message Sets using Message Operators which are visualized as LOS and HOS Engines e.g. DNA being parsed by a Message Transcoder / Operator.

We can define the Virtual Self as a wrapper which contains our SC.

Thinking is when the Self processes Games and Goals

The Self contains the Virtual Self and the Physical Self

Related Video Tutorials

https://archive.org/details/ThinkingMachineDesignVideos

YouTube Playlist

https://www.youtube.com/playlist?list=PLr2wo4sWfjm8EEzgPYgTjA6AeTzJcwJEu

Chapter 1 – Overview of a Being

Here we define the Virtual Person and the Physical person. Later we will apply these ideas to an Android design.

The Window In The Mind

Human consciousness is essentially housed inside the brain's virtual reality system. For example, when light comes into the eye it is upside down. What all biological creatures similar to humans do is that they have genetic code within their brains which creates a virtual copy of what they are seeing. This copy is "synced up" with reality and it is in here that we make decisions on where and how to move. It is a filtered reality where only certain sounds and light wavelengths are considered. Objects are constructed in the Mind based on the vision system such that if one closes ones' eyes one can still see what one was looking at in one's "Minds' Eye". What this means is that our Consciousness actually lives inside this "window in the mind". It is here that we determine how we can solve problems. Therefore to create an "Intelligent Robot" for example, one can create a virtual world for it to move around. Then once this is achieved, all one needs to do is create the hardware to virtualize reality and syncs it up with reality. There is of course a need for some kind of Neural Network. This is why for example that movies work so well in that they mirror the way the brain works by creating another "window" for us to look into similar to what the brain is doing. In the case of a fly, for example, it can also create this internal view of reality so that genetic hardware to create a synchronized view of reality does not require that much physical size. However, the processing and understanding of the objects and the location require extra neural hardware. Next I'll outline the pieces as I see them.

The Inner Window And Inner Self

The self is a self-replicated copy of the body to which all the nerve endings are coupled to. Therefore a human being or any creature even an insect has an inner virtual copy of itself to which the nervous system is coupled and this is essentially who we are. This is the part of use which can feel and taste etc; This inner self continues to exist even if a person for example loses a limb. They are said to have ghost pain. What this is the inner self still existing but no nervous information is passed to this self. The Inner Window is a three dimensional map built up by the brain through which the inner self moves. It is synced up with the external stimuli. The inner self can be placed in any context switched room. Reality is the one which we are in when we are awake. When we sleep the brain context switches us to other rooms which seem "real". They are created by our brain's virtual reality system. In hypnotism, we deal with the person doing the hypnotizing acting as the context switcher and in some cases providing commands to create an inner room in which we inhabit which can appear real.

Tracking Prey

Tracking prey is calculated in the virtual reality view. The key to the design is a fast feedback system to the virtual system via the externally attached sensors. Attack strategy is programmed into the virtual reality system. Typically in lower lifeforms this code/logic is hard coded into the virtual system. In higher life forms the tracking logic employs different strategies. Theoretically a life form can imagine tracking and catching prey while sleeping. All that one needs to do is "create the room" with per-recorded attacks. This is possibly how to have behaviors encoded in DNA. Therefore one needs DNA code that can reverse engineer learning situations into the next generation. So a spider when it is born has a room in its mind which shows it how to build a web kind of like a 3D video tutorial. A baby knows how to suckle because there is a memory showing it how to do it.

DNA

DNA has a timing mechanism. The DNA is itself evolving. Probably the length of the DNA strand map to the time sequence. Behavior based instructions for the inner widow contain instructions for how to behave etc; The DNA contains the complete three dimensional view of the life form in a virtual form independent of the physical form so these is virtual mirroring of the life form created by the DNA.

DNA itself grows and requires an electrical component similar to a machine in order to process signals. Each generation grows the DNA. Therefore the timing of sexual reproduction indicates the genetic code which is passed between a couple. Therefore if this idea is correct the children produced by parents is as much time dependent as it is a part of the random selection process. The DNA is learning while the body is alive. There is a feedback mechanism. Therefore the virtual copy must feed certain information to the DNA in order for an evolutionary style improvement from one generation to the next.

Visual Processing

Typically in Computer Science we use techniques to find edges around shapes. These are crude mechanisms but they work. In the vision section, I will discuss how human vision does it and the types of messaging and mechanisms required.

Typically software uses

Gaussian Blur

Edge detection

Motion detection and building up three-dimensional objects for the Inner window based on movement

Memories

A mind has multiple contexts based inner windows which are three-dimensional rooms. These are context based. Therefore one can see memories as different virtual context switched rooms supported by the visual cortex possibly. Later I will define these as System Consciousness "Rooms".

Semantic Context Switching

The SC needs to be able to context switch depending on the situation. I will cover this is in the Vision Section.

A Rolodex of Memory rooms

One can imagine a Rolodex of memory rooms which contain situation based memories. For example, a baby knows how to swim. Later it does not. So the room are time dependent within the DNA. They are only available at certain times within the DNA life sequence. They may be related to certain sounds/smells or external stimuli such as being in water (context switched + timing).

The Need For Pleasure And Pain

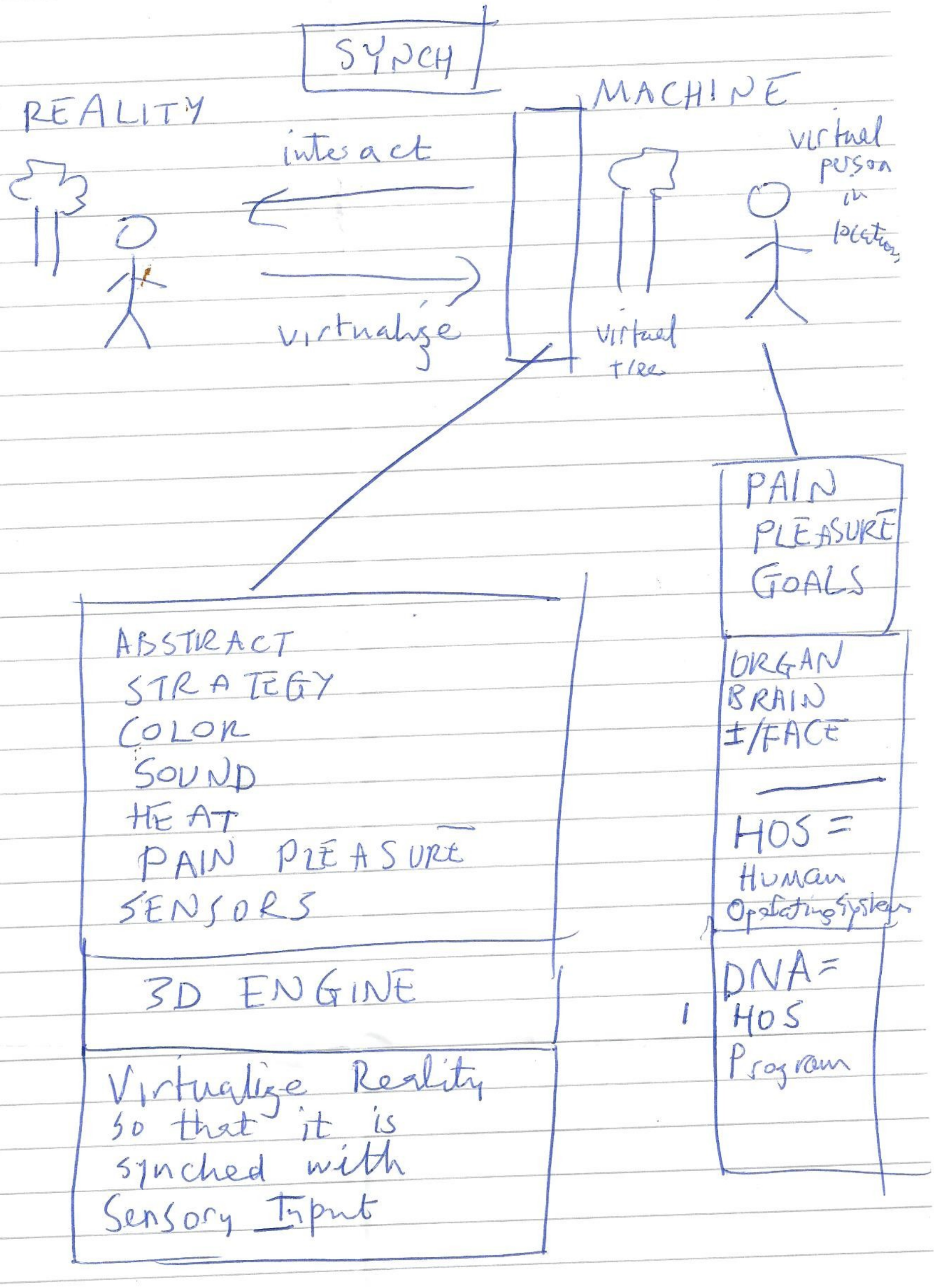
One might be inclined to think that when one is designing an android say that pleasure and pain are optional. The reason these features are required is that pain acts as a natural inhibitor for bad actions. And pleasure acts as a goal driver. One might design an android to get pleasure out of doing a good job for example. If one does not have pleasure and pain then then danger is the machine will be a sociopath. A behavioral inhibitor is also a good idea but grounded rules and some kind of conscience are also good. These are the reasons why humans have these features in my opinion to ensures that social behavior is encouraged at the genetic level.

From a design viewpoint, the virtual person will experience the pain. The skin does not actually feel the pain, it is the virtual person who feels the pain and it is the mind which administers the pain. This is why it can be removed by drugs. It is not there as such and is the product of nerve end stimulation.

HMIC

We can imagine a simple diagram where one has a physical self and a virtual self. The person's HOS has different capabilities and our sense of reality is based on real time synchronization between the virtual and the physical self. Later in the theory, these become MESSAGE TYPES with specialized jobs.

HMIC - Human Machine Interface



ABSTRACT
STRATEGY
COLOR
SOUND
HEAT
PAIN PLEASURE
SENSORS

3D ENGINE

Virtualize Reality so that it is synched with Sensory Input

PAIN
PLEASURE
GOALS

ORGAN
BRAIN
I/FACE

HOS =
Human
Operating System

DNA =
HOS
Program

Human Operating System = HOS , A.OS = Android OS.

DNA is a biological program. Its object oriented in that it can grow and become more complex and its job is to build a being. There is an adaptive mechanism built into it where each new child inherits some information from the parent in addition to the basic physical programming. At the core of the brain is a virtualization mechanism and a mechanism for identifying the shape and form of the being to be built and mapping from the real environment to the virtual in the brain. To get this right requires complex genetic programming which may or may not be evolutionary. It is the best guess of this author that certain aspects of the genetic programming have been initiated by a different type of life form than we are / possibly wave based or nanotech like. This is pure conjecture on the part of the author. However, this does not take away from the overall design itself which can be retrofitted into an android.

Dreaming, What and Why

Using this design pattern for a human being, the way sleep and dreaming works is that the part of the brain we call consciousness is essentially the virtual person in the brain hooked up to reality. Therefore one can imagine the virtual person context switching to another room in the mind. This is the Sleep Room. Here the brain provides the virtual person some form of simple entertainment and stimulation while it does housekeeping tasks such as rebuilding joints and so on. Memories are organized and filed away. The key point is that the person we think of as "I" or "Me" is placed in this other room by the brain and given some time off. The virtual person is I/Me and is context switched into this room. This room is as "Real" as the other room which is built from sensory input and therefore it can be very strange to be in a Dream Sleep which seems completely "Real". All that this means is that the virtual person is context switched into another virtual room.

Context Switching and Remoting

The idea of context switching brings us onto the idea of remoting. If one could map the brain for example and figure out which parts of the brain causes context switching, in theory one could switch a person I/Me into another room which is machine generated or allow remoting into another being/see Engines Under Ursus Sci-Fi/hamster idea where Fowler controls the hamster and makes it do a dance or see movies like Avatar. This is essentially the idea of adding an extra virtual person[s] to a being. An android should support such a facility.

The Nature of Vision

If one moves a camera from left to right such as in a "shaky camera" movie, one cannot see what is to the left or to the right once it goes out of camera view. However, in the human brain one can move one's eyes left and right and the room does not become "shaky". This is because we are each of us positioned inside the virtual room and the moment of our eyes in this room is not the same as what the eyes physically see. The brain fills in the pieces to the left and right and keeps the image stable. This is a particularly effective piece of genetic programming. The brain in effect remembers what is on the left and the right and keeps it filled in to make our experience as seamless as possible. Even if we move our eyes quickly from one side to another there is no disorientation. Try doing this with a camera. Also look to your left so that your nose is in the way of one eye. Notice how the nose becomes transparent. The brain combines the images and ensures you can see through your nose but you can still see it so of. Imagine trying to achieve this effect with two cameras. You'd need some software/special image processing effect.

Imagination

In order to imagine it's often said that left handed people can be more creative. This may be because left handed people can context switch their mind to another virtual "experimental room". Einstein for example could imagine himself sitting on a beam of light in this room. Therefore one can argue that the virtual reality system in our brain is located in the hemisphere where our left handedness is and can context switch more easily.

A Physics Engine Mind

One of the aspects of a mind is the ability to predict and draft ideas. Therefore in the case of an android, it would be useful to base the mind on some kind of a physics engine, hopefully a relativistic one, where the android could predict outcomes accurately in a physical environment.

Blind siding a Mind / Administrator Mode

If one is behind the design of the brain where one makes the assumption (a large one by some thinking) that it was not just evolution that produced this inner window and inner room where a living being such as a human or an android lives, one can include special “blind side” programming which states that certain life forms are invisible to the mind and will be written out. So let’s take a simple example where one designs an android and we put in blind side filters. The filter states: If Android is in the presence of the entity A one becomes simply unaware of it – not just invisible but “unaware”. That is to say that the mind rejects the existence of the being at a programmatic level. This means entity A can simply wander around with impunity in any location and you would be unaware of their presence. This is like an “administrator” or “administrator mode”. So one could just walk into your house and turn on your heating for example and walk right by you and you’d never sense it was there. Probably you’d need some software code to re-target this object to something else. However, probably those who designed the mind would already know about their idea and already have a defense in place. Plus they’d not like anyone trying it. After all they are the administrator and feel like they are the “boss”. This would have to be done let’s say delicately or not at all. I would not recommend trying this personally but the idea is interesting.

Constructing a Virtual Image based on incomplete information

If one arrives in a city and does not look up the virtual copy built must be a best guess based on incomplete information. Therefore, the mind should have templates which one can apply. So if we stand beside a skyscraper and do not look up, we can assume it is the same to the top and place a flat top on it. If we then look up we can update the real virtual image. This is the job of the mind to do this; to create the most realistic image based on the data supplied. Sometime our mind “can play tricks on us” where we think we see something which is not quite there or the mind incorrectly identifies it. We maybe blink and then see something else. This is the mind creating the reality for us. We may be inclined to think we control this. Invariably we do not. The automated part of our brain builds this.

Chapter 2 - DNA And The Building Blocks

The Job of DNA

DNA creates a physical copy of the body and a virtual copy of the body.

The body it builds maps external stimuli into internal stimuli.

The DNA knows what the shape and function of the body parts are.

An Android will also need a Virtual / Physical Descriptor.

Processing Sound

The sound that we head in the Inner Window is manufactured by our Inner Window.

In the same way that images are constructed.

Our inner Mind needs to construct what we think we hear.

A human mind therefore needs the equivalent of a inner sound card or sound perception.

Processing Color

Our inner mind needs to create how to map the wavelengths we see into our concept of Color. I will cover this in Vision.

Sex and Reproduction

If an Android mechanism wants to merge with another, how can this be achieved? Basically we need core code to build any type of virtual shape and function. The sexual reproduction part is covered by a transfer of configuration.

So when two genes compete for example, types of ears for example, this is actually configuration to do with the ear. It is not the code to build a generic ear but the code to the specific changes / alterations

So one can imagine that the start of the DNA is core building an object. Later parts are to do with configuration.

Virtual Image Builder For Vision System

Need a technical solution e.g. an Android app which scans pictures and then converts them into a three dimensional reality in the inner-mind. Something like an OpenGL representation. Home = Inner Mind. Maybe an Android App which you can move around with and it builds a 3D representation of what it sees and turns them into some kind of objects.

Inside this, one places one's self. If this is a drone, then the drone should have a copy of itself in there with rotors and the whole works and degrees of movement. There should be the ability to copy every type of movement and operation in this place so it can "predict" its movement as opposed to just measure where it is at any moment in time. In the case of an android, one should be about to stretch out ones arm and move ones fingers in their virtual place. So there needs to be a version of the self where one is not directly attached to the nerve endings but can control a virtual copy. Then one can "apply" these movements to the actual body or limb.

In the case of an android tablet, then the self is an android table with position and the usual sensors. This could be useful for applications to help the owner of the tablet to ask questions like: How many objects are around me and how far are they away? This is not terribly useful perhaps but one gets the idea.

Reverse Engineering DNA

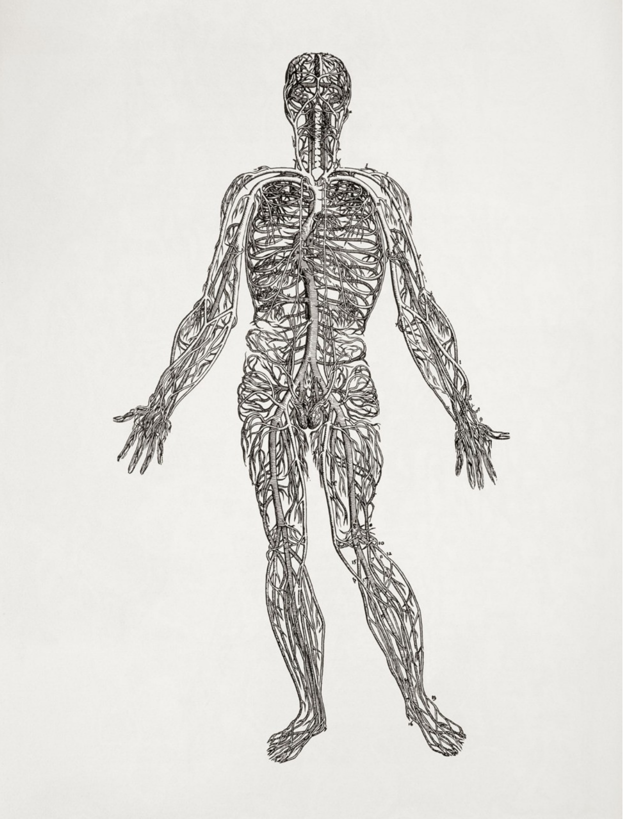
Part of the task of Creating a Thinking Robot is to Reverse Engineer DNA. By understanding how it works, there may be aspects of this which will be useful in the Android Design. This will take a while. However, the idea is to map DNA at a high level as opposed to get caught up in the detail of the Gene Strands. That would be a little like looking at individual lines of code for a larger program. For the purposed of this design, it is assumed that life has been Designed as opposed to just having randomly happened. Evolution is seen as a part of the overall design which is Adaptive Genes, mostly for Geographic Optimization of Genes. The core Brain function will be assumed to have been designed but the nature of the designer will not be discussed but more importantly the design intent.

Improving on the DNA Design

The hope will be to improve upon the DNA design for the Android. It will not be easy to do this but it must be a design intent.

Sensory Nervous System

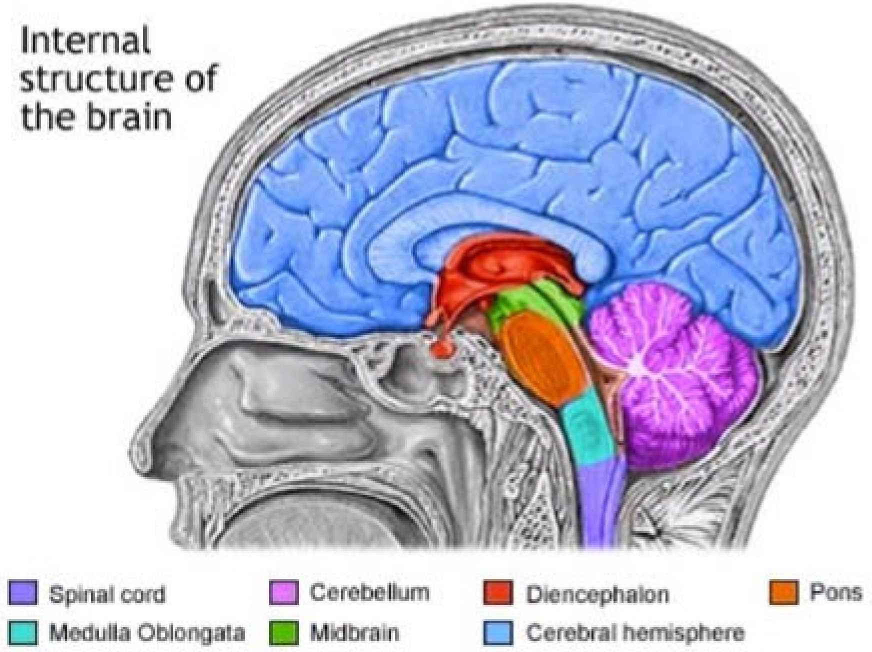
The idea of the nervous system is to send messages to the brain stem where the information is collated to build a real-time system or interaction with the environment. The location and density of nerve endings depend on the requirements. For example in an Android we may want a lot of nerve ending in the hands if it is required to do lot of manual work.



In this design pattern, the nerve ending are mapped to the virtual copy of the being in question.

Where does the Virtual Copy Live?

On Earth, all living beings have some sense of self. Even a fly is self-aware. Therefore in a person the location of the virtual self should be located in the older part of the brain, close to the brain-stem and the modern improvements are build on top of this. Based on an educated guess this is a location near the brain stem called the Pons. The exact location is not important but there should be a location where the Virtual Self is context switched in and out. The guess is based on the fact that it is the older part of the brain and is responsible for sleeping and awake states which require some kind of context switching based on the design I have described already where one switches virtual rooms.



Note how the Pons is near the Cerebellum which controls motor function and is also close to the spinal chord where muscle messages can be sent to the body of the person. We can almost think of the brain as a Layered Architecture where the base functions are close to the brain stem and the mammalian functions have been added later.

Also the Vision processing runs along the optic chord to the base of the brain near where the Pons and Cerebellum is so that that virtual copy can be created and quickly receive real time information.

The optic nerve runs into the diencephalon so it's like there is some kind of pattern recognition system near here.

Therefore it makes sense for an Android to co-locate the Virtual copy close to core motor functions where context switching can happen faster using a message based architecture.

Lack of Coordination versus Being Athletic

Often people are referred to as lacking coordination. What this means is that the mapping between the virtual self and physical self are not perfectly synced. For example, one might imagine in one's virtual world that moving an object in a certain way would mean it would go in that direction but in reality it does not do this. This is because the inner copy does not have an accurate copy of the external world in its internal world.

Alternatively, one is called Athletic when one can kick a ball for example and know where the ball will end up either to another player or in the net of the opponent. In these case, the virtual copy contains a very accurate and synced copy between the real world and the virtual world.

So an Android should have

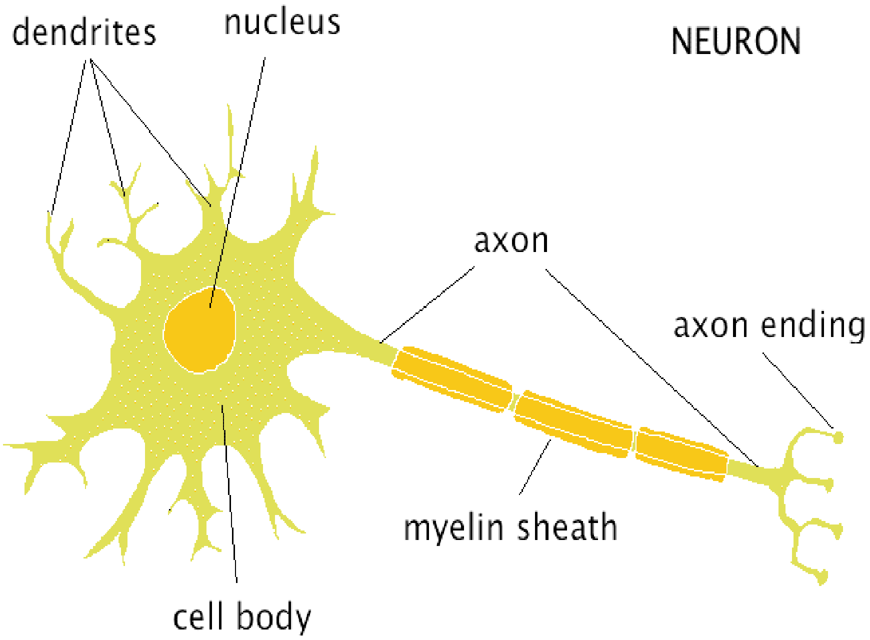
1. Fast real time updating of sensory input
2. Accurate modeling of the real world in the virtual copy
3. Ability to accurately predict movement and results in real time
4. For this, as stated before, an accurate predictive Physics Engine in the Virtual copy

Structure Versus Messaging

The high level design goal is to produce intelligence from several distinct components.

1. A message based system
2. A physical structure through which the messages flow
3. A structure which supports receiving and emitting message types
4. Specialized actors whose messaging logic determines their type.

Neurons



We consider a Neuron as a message receiver and transmitter whose structure stores some kind of message state.

For example, image based Neurons are different than Hearing Neurons

Dendrite Collect Electrical Signals. This is the Actor which receives messages from connected Actors.

The Nucleus / Cell Body contains the Message Processing Logic and knows what type of Neuron it is e.g. Image Neuron.

The Axon Sends the outgoing message to other Actors or to an effector cell e.g. muscle, gland or organ cell.

Building The Structure – The Job Of The DNA / RNA

The jobs are

1. Build the body iteratively (what we call growing)

2. Operate the body (this is the job of the individual cells and the master processes)

So we need a messaging solution for

1. Growing the body
2. Operating the body

Theoretically an Android does not need to be grown but rather it is assembled in a factory. However, it would be better if the robots or even folks who assemble the Androids adhere to some kind of messaging protocol. This would mean that if we want to update an android we may not need to totally throw it away.

For example

Message: Attach Arm, Detach Arm, Re/build Arm

If an Android adheres to this protocol in its structure then if we want to upgrade the arm, the robot knows how to do it itself once we send it the message, rather than an operator come in an remove a bolt in some specified location based on a manufacturer booklet.

Certain animals for example can regrow a limb because their DNA supports these messages. For example a deer can regrow an Antler and a Mexican Axolotl can regrow limbs. There is no good reason why an Android should not be able to regrow itself if that is how it is constructed.

An Android should also die which would just mean that it has a message called "Deconstruct". This means that it can just take itself apart and certain components reusable.

However if we assume it's "assembled" then we run into trouble so we should create a build in set of "Grow Myself Messages" and "Regrow parts of me" messages and even "Fall apart".

In many ways this is how we work, we know how to grow ourselves the messages are in our Cell DNA. When we die, our body totally falls apart and all of our cellular material is reusable.

So we could image a robot “Core” entering a factory and it issues commands to the build assembly telling it how it should be assembled. At a certain point, it assembles itself!

If one thinks about how a baby is created, the embryo tells the body what it needs and to a certain extent the baby is in charge of the mother until it reaches a certain period of development and then the baby for the most part grows itself.

The umbilical chord is the message passing system between the mother and the child.

Birth is when the child leaves the “factory” / womb which is the mother.

In the case of an Android one could think of a “mother” factory where the mother builds the child Android but tells the mother what it needs. In a human case the mother transfers some food and nutritional information based on the location.

Typically we see this where a child has certain food preferences based on the diet of the parent. This is achieved by a messaging system.

Each Actor Contains the entire program

If we think of a Cell as a Actor, then each cell contains the whole program. This is analogous to a Singleton in programming terms which all cells own/share. So we can think of the entire program for a being as being the DNA.

Therefore each Actor is created by some kind of message which indicates its type and purpose.

As I've stated already, each Cell/Actor need to contain timing information as to how far they are in the program. Typically we call this process Aging in our case. In the case of an Android, we might prefer to keep the aging process to a minimum but aging even for an Android might be a good idea as the next model could learn from its poor design decisions and build “Improvement” messages for its next theoretical self build when it deconstructs and reconstructs. Therefore we could think of this as some kind of evolutionary process.

We might think of this as altering the “Android Rebuild Code”.

The Tricky Part - Building The Virtual And The Physical

If one wishes to copy the human approach one has a pretty tricky task. That is, design a messaging system which will

1. Specify a being with predetermined behavior and function
2. Build a Virtual “Self” Copy of a being from scratch
3. Build a Physical Copy of said being
4. Design a system which syncs both versions as it develops
5. Permit the Self to change shape, function and behavior over time
6. Defend from external attacks either microscopic or macroscopic
7. Base the being on locally sourced materials
8. Regulate temperature and energy needs
9. Rebuild when stuff breaks and identify internal problems

There are many other requirements but this all has to be achieved by DNA generating a messaging system. The first four requirements are probably the starting point. Therefore, the start of the DNA strand should specify some kind of locale information and then have behavior and function stated at a high level.

e.g.

MAYBE LIKE A STEM CELL

NEED EARTH CONDITIONS LISTED

CARBON BASED AND LIST OF OTHER BITS/ELEMENTS

GETTER MESSAGES FOR RETRIEVING THESE COMPONENTS

MAKE FIRST CELL

Every cell will have something like

I AM HERE IN DNA PROGRAM

THIS IS THE SYSTEM TIME

THIS IS MY TIME

THIS IS WHAT I DO

THESE ARE THE MESSAGES I SEND

THESE ARE THE MESSAGES I RECEIVE

THIS IS HOW AND WHEN I REPLICATE

THIS IS HOW I FINISH

High Degrees of Message Passing Are The Key To Intelligence

Ants have only a few chromosomes (one or two) and have limited intelligence but when they work in groups they exhibit intelligent behavior. So what this tells us is that message passing is the key to intelligence. Ants do this by means of scent messaging and therefore the intelligence is transferred to the group. In mammals, the message passing is primarily in the skull and not externalized. There may of course be some herd behavior. So an intelligent Android will have a high degree of message passing in its brain area. It would also be useful for Android intelligence to be "Clustered" so that it could also behave intelligently in a group manner like an Ant colony having the best of both worlds. Scaling intelligence "out" and "up".

Protecting Core Systems – An Onion Model

Although certain cells can send messages to do with damage and regeneration, an Android where possible should adopt an onion model where the core part is the most protected. For example, in the human brain outer neurons can be regrown. However, this will incur the loss of memories and some function which may be relearned. So, the brain-stem contains the core motor functions and operations like telling the heart to keep beating. Therefore it makes sense for an Android to place all its core functions close to the center of the unit in the same way that the brain-stem houses all the critical systems of the skull.

Alternatively using a Hive approach one can share memories across Androids or in contemporary parlance stored in the Cloud which is the contemporary Hive name.

The Purpose of Chromosomes And Strategy Deltas

In this theory, the Chromosomes form the program to build the living thing. The number of Chromosomes determines the number of deltas the life-form has undergone to the present time. Evolution inside the Chromosome is characterized by more DNA where the life-form adopts different survival strategies.

Therefore we need a survival strategy message which converts ideas to DNA

e.g. MAP APPROACH to DNA

In humans, typically this is done by Natural Selection based on paired approaches.

However, this is not the only way that this can be done.

For example, when an intruder enters a Japanese Honeybee nest, the drones send each other signals to raise their body temperature and cook the intruder. This is an agreed upon behavior via Hive messaging and a Hive strategy.

This requires an idea to be encoded in the DNA and shared among each Hive generational.

An Android should also be able to encode survival strategies into its A.DNA = Android DNA which it can share with other Androids.

Thousands of letters in DNA/messages

22 + X/Y Chromosomes

Approximately 2000 genes per chromosome, millions of base pairs

So a gene is a message sentence e.g. move salt particles from a to b. This appears to be a biological programming language with structures, timers, looping mechanisms for generating multiple protein types from the same core code and advanced messaging. It's possible that DNA/Chromosomes are built on top of a lower level API.

Letter combinations forms a message

Knowledge of Three-dimensional Building Blocks And Properties of Materials

Apart from the advanced messaging and timing mechanisms in DNA, it also has a knowledge of protein types and their behaviors. An android therefore needs to know how things fits together physically and what they do. For example, if you add more devices you need a larger power system. Therefore the messaging system needs Planner Messaging.

e.g.

ADD DEVICE A NEEDS POWER X

ADD DEVICE B NEEDS POWER Y

ADD POWER UNIT NEEDS TO SUPPORT X + Y

In the same way in a human, if you become larger you must ensure the heart is sufficient.

The sequencing is also important. Apart from the evolutionary aspect of growth, there is a need to produce planning steps. For example, ones needs to build certain bits first.

BUILD A

BUILD B

BUILD C

This must all be done to a master blue print. Messages need to be sent to the planner cells/actors to tell them when something is done.

It's quite a good idea also to build in some kind of test framework for each step, such as

BUILD A

TEST A (VIRTUAL TO PHYSICAL POSSIBLY)

FAIL END BUILD

In the case of organic life, one sees a child moving inside a parent, trying out moves and appearing to do some kind of thinking when it has a certain size.

Also some of the building blocks should be reusable like in DNA for building various protein types using common code.

MOVE A

ASSEMBLE <TYPE>

Mapping Conscious Messaging To System Messaging

If we imagine a human being has a HOS (Human Operating System) one part of this is SC or System Consciousness. This is what we are. We are the part of the system which deals with hunter-gathering scenarios. However, when the body runs out of food, it communicates with SC and the message is

(HOS) WATER LOW → (SC) I AM THIRSTY

(HOS) PROTEIN LOW → (SC) I AM HUNGRY

Does an Android need this? Not really but the reason it is there I think is not to overwhelm Consciousness with too many system alerts and warning.

Cell Messaging Using Dynamic Messages and Message Algebra

DNA cells individually use message passing. The key to cell messaging is to use dynamic message types. This is called an Untyped Actor. This leads to the idea of Message Algebra where we solve problems like Pattern Recognition, Auditory Processing and Language Processing with Message Algebra.

A typical pattern recognition problem is to identify sounds or visual letters and match them to a learned value.

Therefore we can imagine a human being using Message Algebra as the basis for logic and reason.

In fact we can extend this idea to the point where we imagine that thoughts are essentially message algebra which are localized to a person's language.

If one never learned a language, the natural message algebra in our brains would produce some kind of language.

Therefore

language → message algebra

vision → message algebra

auditory processing → message algebra

message algebra → logic and reason

logic and reason → message algebra

So the challenge is to produce message algebra from cells containing information and form logic and reason. Therefore a simple example is, find the letters on the page. Then figure out the words and the sentences. Then answer the question. Then generate a response. Do this all just using message algebra.

The first task must be to turn everything into message algebra, no matter the source.

This is the approach an Android must take when problem solving.

Chapter 3 - Message Algebra Requirements

Next we'll look into a message algebra for the Cells and initially define what they do and how they can work.

Messages should have the ability to convey

1. Logic and reason, thoughts
2. Geometry
3. Three-dimensional concepts such as movement, distance and so on
4. Object description and composition, location
5. Problem solving
6. Personal state such as emotions and desires
7. Physical state such as pain, pleasure and so on
8. External state such temperature, light and so on
9. Ability to maps some of this message algebra back to physical systems such as voice, movement and so on

Basically anything to with what we think of as a living being.

This can be covered under Human Operating System as the HOS Inner Language

e.g. These are all message

THOUGHT I HAVE A HEAD COLD, TIMEOUT WAITING FOR IT TO GO AWAY

THOUGHT SOLVE I NEED TO GO TO THE DOCTOR

EYES LOOKUP DOCTOR

MOVEMENT GET UP

SOLVE I AM HERE DOCTOR IS THERE

FIND ROUTE IN VIRTUAL MEMORY LOCATION

MOVEMENT CAR DRIVE

The Reason For Language

The sounds and the grammar are just a mechanism for one human to communicate these shared internal message types to others using a common framework. There are no fixed rules for the best way to do this therefore there are many language types and approaches, however all can be mapped back to the brain's HOS Language.

So

ALL HUMAN LANGUAGES → HOS Internal Language

Certain languages have a wider range of expressions but all pretty much map the same HOS messages. The vocabulary and grammar are actually incidental even though in written text many folks get worked up about things like punctuation and so on but the most important thing being messaged are the shared HOS Messages.

JOHN FEELS HOT

SUN BRIGHT

LOCATION BEACH

WALKING

John took a casual stroll on the beach and felt hot

The blazing sun beamed down

What is Consciousness?

In this design pattern, Consciousness is the part of the HOS (SC) where messages are directed (Internal Filtered and External Filtered). There are a wide range of types and they connect to a Learning part of the HOS where one gains Experience. What experience means is that groups of Messages are gathered together under a single Learning Message so that they can be recalled at a later stage. The learning mechanisms which is part of Consciousness matches Messages and gathers them together as an Optimization.

e.g.

LEARNING HOW TO BUILD TABLE

ASSEMBLE LEGS

ASSEMBLE TABLE TOP

CONNECT LEGS TO TABLE TOP

TASK COMPLETE

STORE IN MEMORY

What Is Memory?

Memory is a place where Message are stored and triggered based on similar message patterns arriving into Consciousness.

e.g.

ARRIVE IN LOCATION

SPOT SOME LAND MARKS

MESSAGES ARE PASSED THROUGH MEMORY

TRIGGERS A MEMORY OF A PREVIOUS VISIT

VIDEO MESSAGES FIRED

CONSCIOUSNESS RECALLS LAST VISIT

Note: Consciousness can rebuild a place from memory based on VISUAL MESSAGES

e.g.

VISUAL MEMORY HOUSE

VISUAL MEMORY KITCHEN

VISUAL MEMORY MEAL

These messages are routed to the Virtual System and can build an Image where one can place ones self. Consciousness can then context switch the Self back to the real time virtual view of reality.

A Simple Example – Matching some letter and Learning It

//ALREADY LEARNED

Break image into dots which are actors

Generate VISUAL ALGEBRA messages e.g. for letter A

VISUAL MATCH POINTING UP LINES COMING TOGETHER AT ONE END

VISUAL MATCH LINE ACROSS IN MIDDLE

PASS MESSAGES TO CONSCIOUSNESS

TIGGER MEMORY MESSAGE MATCH

TRIGGER AUDIO A SOUND – A

MAP MESSAGE A TO OUTPUT A

Person says "A"

TRIGGER LEARNING MEESSAGE – A SYMBOL

Image is mapped to Sound and Image

Affecting the Message Passing

LSD affects the message passing in a human body and the virtual room that is built is essentially faulty. This is already known. They key point to make here is that this is simple proof that it's not just the case that light ends up in our eyes. We rebuild virtual rooms based on these video and object messages. For example, a person can imagine that a chair is floating or that walls bend shape or light distorts. This is called "tripping" but what it really means is that the message passing is altered by the drug in question.

Theoretically from a Science-Fiction perspective one could be jacked-into another computer generated room if one could understand the messaging protocol. This also means a person's consciousness can be parked in another part of its brain which would not be a good thing so this type of work must be done under ethical supervision "permanent sleep". Also, some books talk about multiple personality syndrome. According to this theory this means someone has more than one virtual Self inside ones brain who can be context switched in and out depending on certain triggers.

Also those with brain illnesses can be traced using this design pattern to have faulty message generation. Therefore if it can be ascertained as to what these messages are, then in theory the condition is treatable in a more targeted way.

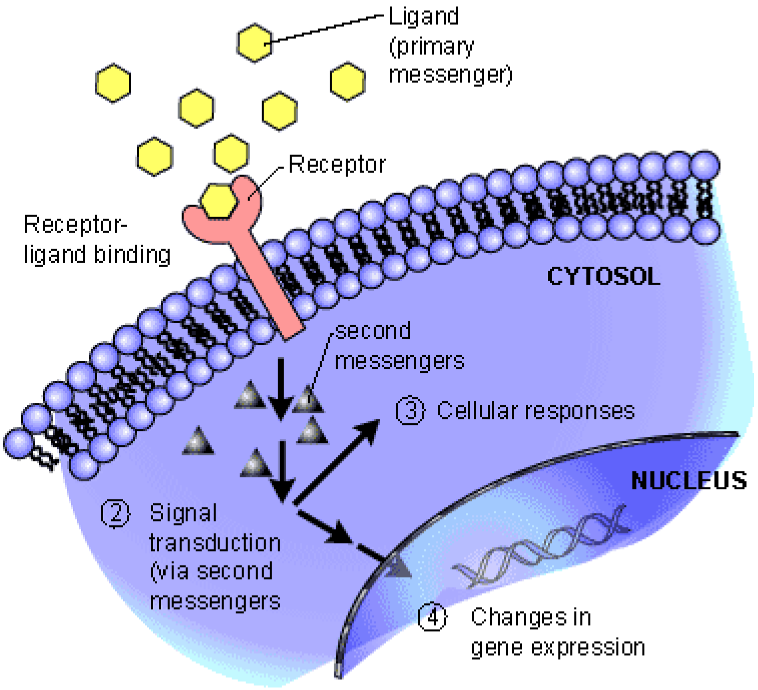
Presently, drugs are administered to lessen thought / message generation. Therefore the subject in question ends up in a zombie-like state where even walking is slowed down which is another message passing mechanism.

So the benefits are clear in reverse engineering the message algebra; the ability to treat and understand illness. Also if one can speed up message passing in a healthy way then one can increase IQ and combat abilities.

It's unclear if one wants an Android to experience a high some people seek out. I think it's more of an unintended consequence of our brain chemistry than a design decision at the DNA level. The body / DNA does however appear to punish those who achieve these highs. For example, alcohol produces hang overs, drug addicts become junkies and so on. So I don't think an Android design should permit highs but maybe if Androids become real, maybe they will seek them out themselves. Who knows? Android junkies... now there's a thought.

What do Cell Receptors Look Like?

The receptor sends and receives the message I refer to. The android untyped actors must copy these. Inside is the Nucleus which is the core DNA program so to speak. More than likely the Cytosol contains run time information / state of the cell in question.



Aggregating The Message Information

The purpose of System Consciousness is to aggregate the incoming data from all the data sources. These include vision, sound, touch and so on. The message all come into SC at roughly the same time. However vision processes more slowly and is more accurate with regards to positional information. Therefore the messages are like

CHAIR AT POSITION X,Y,Z

PERSON SITTING ON CHAIR

It takes a certain amount of time to figure out it's a chair and there is a person who one might know. Sound on the other hand takes less processing so looking at the mouth of the person, one might be able to deduce who the person is based on the voice.

VOICE FRED

All of this information goes from the SC into the Visual Cortex which builds the room and places the person on the seat.

Note that because we are using a messaging mechanism the Audio messages do not have to come into the SC at the same time as the Visual information.

SC uses the “Best Guess” algorithm which means that it builds a real time information based on the Messages it gets at any moment in time.

One might be inclined to think we really are seeing what is in front of us but the old lady, young lady example shows us that we are trying to figure out what we are seeing via messaging.

So when you see an old lady, the SC gets a message

OLD LADY

and vision builds that

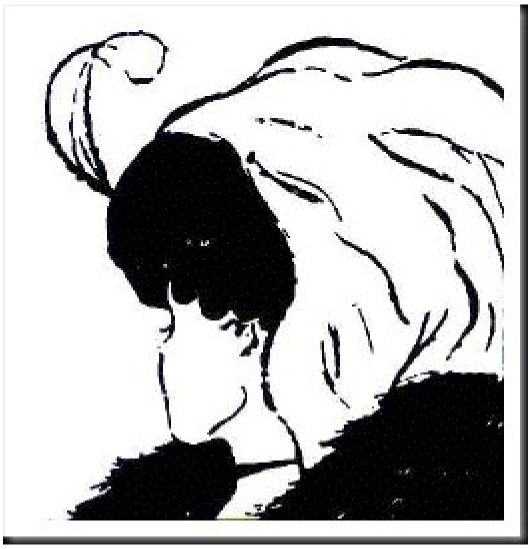
Then when you see the young lady your SC gets the message

YOUNG LADY

and suddenly the other image disappears by means of context switching

To switch back requires some concentration and then one can send OLD LADY to the vision processing system.

It's hard to see both at the same time as they occupy the same location and the brain visual builder has trouble with this but it can be done.



This message has semantic context in the brain where the message YOUNG LADY also include

SHOES

FASHION

FEATHER

COAT

and so on which is how we match.

So the brain matches certain images to messages and they produce a match in brain memory for either OLD LADY or YOUNG LADY.

In my case, once I see the LONG NOSE and EYE, then I see OLD LADY

If I see HAIR and COAT and FEATHER I see YOUNG LADY.

In order to see FEATHER, the vision actors need to match on this part of the image and produce the message FEATHER at X,Y.

Note that sometimes the SC best guess can “trick us”. All that this really means is that the image that we see is not what is really generated but it's the best guess by the brain. The example I saw was a flashing image with two clicks. To the viewer it seems like there were two flashing images but there was really only one. When the sound was removed we just saw one flash.

So what we have is (First test case, two clicks and one flash simultaneously)

--Message arrive close together within a second or so

SOUND CLICK

SOUND CLICK

IMAGE FLASH

We can think of Sound messages coming in on one queue and Image messages coming in one a image queue with per second thresholds. Then they need to be aggregated into a result per second for the Visual system.

Best guess by brain then produces result for visual system

SOUND CLICK

IMAGE FLASH

..

SOUND CLICK

IMAGE FLASH

---> send to vision system and build virtual view for Self

Person counts/sees two flashes when there was only one

Drop the sound and all we get is

IMAGE FLASH

So we count one

This is because the Vision Messaging algebra is not totally in sync with the sound messages so it's a best guess.

This is essentially “Aggregating Messaging Algebra from message queues over discrete time frames”

So the SC is sampling Messages every second or so and has a threshold.

Theoretical Basis For Cancer And Theoretical Solution

In this design pattern a cell is a state based machine which contains the entire program for every cell. Using this design pattern the Cytosol is the part of the cell which sends and receives messages. It also contains a timer and an understanding of what types of messages to send and receive based on the current state of the Cell type. Basically, the Cells knows how old it is and what to do next based on the messaging it sends and receives. One operation a Cell can do is issue a command to grow a new one. Normally this works fine when the DNA programs is healthy.

Let's take a theoretical example of a hereditary disease like Breast Cancer.

This is passed from parent to child. At a specific time in the life of the person with this cancer, it activates mysteriously and becomes rapidly lethal.

Let's try and understand how this works theoretically. Note: I approach this problem purely from a Theoretical viewpoint. I have no training in this area but I apply pure theoretical logic!

First

1. Every cell has a timer, let's not concern ourself with how for now
2. Each cell knows the next cellular action to take at a cellular level based on the time and the last message
3. So in this type of Cancer the cell **timer** and **last message received** produces a **faulty/deadly response** for the person with cells of this type
4. It issues a command to generate cancerous cells
5. However the good news is that the cell worked perfectly for many years before this so the solution is to reset the cellular timer to a point where it behaves correctly.

So for this type of Cancer, the theoretical solution is to use a gene therapy that figures out how Cells count time and reset it by several years.

Therefore the Cancerous cell in this type of case never enters the state which tells the Cell to manufacture cancer cells.

This is the theoretical solution for this case and would need to be tested for validity by professionals in this area of study.

Note: There are other environmental impacts which can cause cell messaging to get confused and is not covered in this example e.g. smoking, high EM fields, radiation exposure etc;

Obviously in the case where the Cells DNA has been damaged due to environmental conditions the approach that must be taken here is to issue a CELL DECEASE message for the damaged ones only. This would require a DNA therapy which checks the DNA program of each cell for validity.

Something like

```
EXAMINE DNA
CHECK FOR DEFECTS
IF DEFECTIVE, ISSUE CELL SHUTDOWN MESSAGE
```

An Android should also have this type of defensive mechanism built into its A.DNA as well if that is the approach taken.

DNA, Messaging and a Molecular Knowledge Base

In order to build a three-dimensional living being, one needs not only DNA as the master program but one also needs messaging and a molecular knowledge base.

The molecular knowledge base needs to be in the base code as this is the mechanism which all living things use. This is not just the four DNA molecules, this is a database of the molecules on the planet the DNA has found itself probably when it first "mines" the world for suitability. More than likely, these molecule types are common in the universe but on certain planets, the base building blocks may be present on the planet in question. Personally I think it's a mistake to think that only worlds which have water can build a DNA structure so the answer lies in the base DNA which probably includes some kind of search protocol for the base molecules available on the planet in question before starting.

There also needs to be large sections in the DNA which match up Cells types with their associated run time messaging protocols. Therefore DNA should have sub-routines which accept different message types. A simple example is a gene which builds proteins. This code has been found to be reusable, therefore we can conclude that it accepts many different types of message builders

e.g.

BUILD PROTEIN A

BUILD PROTEIN B

This is a standard computer science approach to re-use a module which has common code. Therefore the DNA is most likely highly reusable, so the key here is to match the message types to the DBA sub-routine.

Also it can be concluded that the messaging is almost as important if not more important than the DNA routines itself in some cases.

So an Android must have a Molecular Database, Messaging Protocols and re-usable builder Actors.

Message Work flow for Cravings aka High Priority Messaging

Typical simplified messaging work flow here is

Mother has Craving

Fetus messages mother

1. I NEED X WITHIN Y MINUTES – HIGH PRIORITY

2. MESSAGE PASSED TO MOTHER
3. HOS TRANSFERS MESSAGE TO BRAIN-STEM
4. MESSAGE IS PASSED TO SC (SYSTEM CONSCIOUSNESS)
5. X IS LOOKED UP IN SC FOR MATCHED FOOD TYPES WITH HIGH PRIORITY
6. SC FINDS A COMPATIBLE FOOD AND GENERATES MESSAGE
7. I NEED FOOD A
8. MOTHER GETS CRAVING FOR FOOD A

This type of messaging behavior is probably present on the XY gene.

An Android should probably support High Priority messaging and also the ability to map message types to materials.

e.g.

I NEED X

LOCATE Y

LOCATE Z

Learning About Ones Environment And Dynamic Messaging

It's been shown through human studies that an allergy can be avoided in a child if the mother consumes certain types of foods. In this use case, one can imagine that DNA contains the code and the messages which are required to build the child in question. However, there may also be environmental information which should be passed to the child. In particular, the types of foods readily available in the current location.

So the forming child can be messaged about its environment while it is forming to make sure it is more adapted to that environment. The idea here is that the DNA accepts locale information while it is forming.

LOCALE FOOD STUFF X

LOCALE HANDLE FOOD X

LOCALE FOOD X DETAILS Y,Z

One can imagine that this type of messaging is sent to the child while it is forming.

Therefore an Android when it is being prepared to be release into the “real world” it should be sent LOCALE information like

LOCALE POWER SUPPLY 110V

LOCALE CITY LOCATION X

LOCALE COUNTRY Y

LOCALE LANGUAGE Z

Endocrinology And The Need For Slow / General Messaging

We have a fast messaging system that is part of either the HOS or controlled by the SC. For example, HOS controls heart-beat and SC controls vision and hearing for example. If we run, then our heart beats faster because of the HOS, we can however choose to run or not to. Now, these systems are quick response.   If you are chased by a tiger, you need fast internal response time.

In the human body however, we have many specialized areas which can reach thresholds but not fail immediately. For example, let's say the HOS gets reports from some cells

I AM LOW ON GLUCOSE

This message goes to the brain and it is routed to the Endocrine system where it send hormone messages to the various organs to produce more GLUCOSE. The body needs to act on these messages. One would expect that if a message occurs constantly over a time period them the endocrine system will report something like HOS THRESHOLD REACHED

FEELING UNWELL with a higher priority. If the threshold is not reached it will report something like.

PRODCUDE GLUCOSE

BURN FAT

EAT SOMETHING

The body can also find it has too much GLCOSE

I HAVE TOO MUCH GLUCOSE

In this case, a message it sent to

LOWER GLUCOSE

PRODUCE INSULIN

The key point to make here is that it takes a certain amount of time for different parts of the body to execute upon these commands and complete them. Typically, there are many specialized cells with many types of message support to finish the initial goal message. Also, they are not connected by means of nerve endings.

So the solution is to send messages with low response times into the circulatory system (minutes to hours to days). This type of message typically needs some kind of routing information because it is traveling on a general transport route. Typically, the nervous system handles the routing implicitly.

ENDOC MESSAGE LOW PRIO LOWER GLUCOSE ORGAN X

This is send into the circulatory system. So we can see the circulatory system as a kind of message bus like a modern highway.

Therefore, an Android should also support a general message bus between all the major components in it. Typically in a PC architecture we have a message bus so it needs to be something like this.

A typical message might be

POWER CONSUMPTION NEARING THRESHOLD, RECHARGE SOON, ROUTE TO A.OS

This message does not have to be acted upon immediately but in a reasonable time frame. A charge point needs to be located.

Understanding both Electricity And Molecular Connections

In the fast Neuron based messaging, the DNA takes advantage of its knowledge of electricity. For example, the Neurons fire and boost electrical charge using certain chemical components which include ions.

An Android should also have this chemical understanding built into it even if it does not need it there and then. For example, it may need to build some kind of basic electric generator using Cells or Solar technology possibly.

POWER LOW

NO ELECTRICAL POWER POINT

ASSEMBLE GENERATOR X

How Do Cells Know What The Messages Mean?

The DNA strands contain blocks of code which build the cells and include the supported messages. For example kidney cells support different messaging than heart cells.

Also, a full copy of the DNA exists in each cell so all Cells know how to generate a message from one place / organ to another and under what conditions to generate those messages.

In an Android, the different part of the machine should support Actor types which support their specialized messaging protocols. Plus they should also know all messages.

What does it mean for something to be Sentient?

In this architecture Sentience is when the A.OS has a Messaging Algebra which is sufficiently developed that it matches a human beings own DNA based HOS messaging. By this I mean that the range and scope of messaging in the A.OS is able to do things like problem solving, language processing and so on so that a human thinks that the machine "exhibits intelligence". However, what this means according to this architecture is that the machine matches the Message Algebra capabilities of the human being or life form in question. Typically this would be defined by the Turing Test or something similar. Therefore according to this approach widening the message algebra of a living being increases its intelligence. Obviously there need to be accompanying cell support and the overall system needs to be useful.

Adaptive Evolution and Adaptive Messaging

In this theory contrary to Darwinian Evolution, Adaptive Evolution is achieved by means of the Adaptive Messaging Algebra. Typically in Darwinian Evolution some environmental factor comes into play and causes a change in the characteristic of a biological being.

In this theory, Adaptive Messages can be sent from the living being to itself and its reproductive system so that it knows what is working and what is not.

For example

ADAPTIVE MESSAGE FINGERS TOO SHORT

Therefore when the being reproduces some of these messages are sent to reproductive system and there is a chance that for example, the Sperm will implement adaptive message changes to DNA for the next generation.

The adaptive message can be mapped to DNA attributes. Therefore the **"being knows best"** design pattern in the same way that a being knows its preferred mate without being told.

Example

If a fly is red and all the flora and fauna are purple, it will send an adaptive message

ADAPTIVE MESSAGE PURPLE IS SAFER

It's up to the DNA to re-code this so that the next generation of the creature fits into its surroundings better. Note: Only a certain percentage of those born will contain this preferred improvement. This is how this theory describes **adaptive evolution**.

Therefore, using this approach an Android should be actively using ADAPTIVE MESSAGING to improve its ability to fit into its environment for itself and future android designs.

Example

Two samples creatures left after dinosaurs

A. One likes to dig for food

B. The other likes to pick berries off branches

Both send adaptive messages

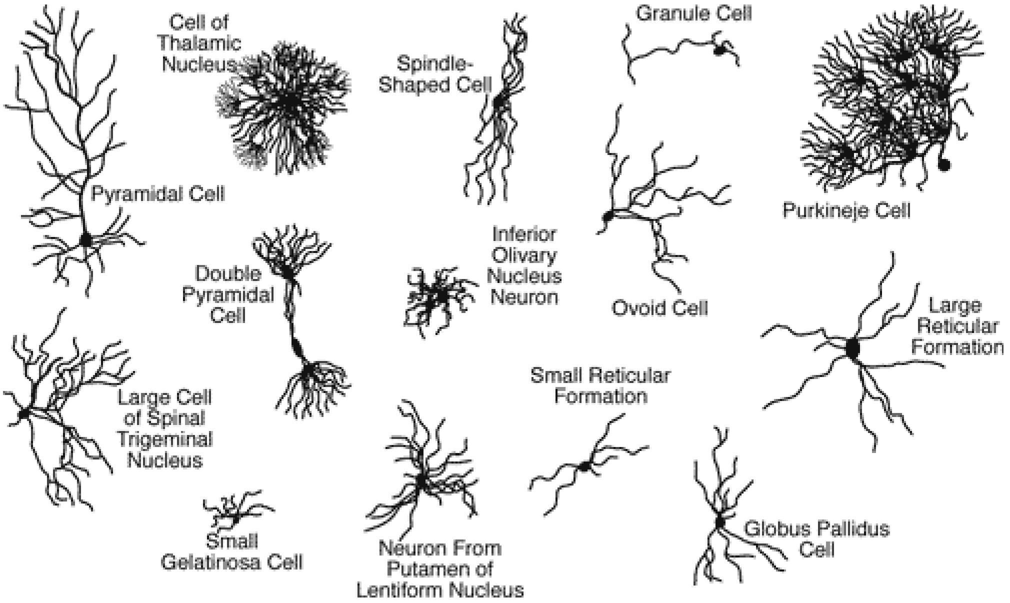
A. (Adaptive Message) Make claws bigger for digging. → In the long term this becomes a hedgehog or maybe even a mouse.

B. (Adaptive Message) Make neck longer for berries. → In the longer term this becomes a giraffe.

Building a messaging Algebra AND,OR,NOT (SQL like lookups)

There are different types of Neurons in the brain.

Inter neuron, Sensory Neuron, Motor neuron etc;


Pyramidal Cell
Cell of Thalamic Nucleus
Spindle-Shaped Cell
Granule Cell
Purkineje Cell
Double Pyramidal Cell
Inferior Olivary Nucleus Neuron
Ovoid Cell
Large Reticular Formation
Large Cell of Spinal Trigeminal Nucleus
Small Reticular Formation
Small Gelatinosa Cell
Neuron From Putamen of Lentiform Nucleus
Globus Pallidus Cell

Instead of seeing these as **Neuron Types**, we can see these as **Message Algebra types**.

Some send and receive messages to the Nervous system.

Others talk to muscles and so on.

However for intelligent behavior in the SC, we need there to be some kind of a lookup language which supports AND, OR, NOT.

From the earlier image example (YOUNG LADY versus OLD LADY)

FIND MESSAGES RELATED TO FEATHER, SHOES, FASHION, FEMALE FACE

this returns messages to the SC

YOUNG LADY

TREND SETTING

COOL

LINKS TO FASHION IMAGES

and so on.

From this list we further refine until we find the best answer which is YOUNG LADY.

There are Messages Neurons to do this according to this design pattern for life.

However we also need the concept of AND, NOT and OR

When we solve logic problems, these operations come logically to us and it is the belief of this theory that our Neurons also contain these MESSAGE ALGEBRA TYPES

FIND MESSAGES WHICH HAVE SHOES WITH NO HIGH HEELS AND (LEATHER OR SUEDE)

A list is returned from the Neural net memory

SHOE TYPE A

SHOE TYPE B

The SC will continuously refine the results with known MESSAGE INFOFORMATION until it is narrowed down to a BEST GUESS.

SC KNOWN INPUT MESSAGES

LOOKUP RELATED RESULTS

FIND RELATED RESULTS UNTIL CLOSEST MATCH FOUND

Using this approach we find semantic content which most closely matches the MESSAGE we receive.

Therefore an Android should have actors with many to many relationships and a message based algorithm for searching the SC Memory centers for related Messages most closely matching the input messages. Remember that everything is a message in this architecture and different actor types hold different messages.

Joining Message Queues

The SC has an architecture where different message groups are time sliced. For example if the sample rate is 10 milliseconds for input exaggerator, we could get many many sounds and just one image. We may need to pair up/ group messages, for example pairing sounds with images. Therefore, the Message aggregator which talks with the Virtual Image Builder makes a best guess at building the view for SC Virtual Self with paired (SOUND,IMAGE) if they are available.

How do we learn messages?

Everything we do is turned into messages and stored in our Message memory. Typically only a certain percentage is "learned" on each pass which means stored as messages.

For an Android we would hope to learn all of the content on the first pass.

ADD MESSAGE X RELATES TO Y,Z

Learning / Purpose of sleep – gathering ones thoughts

At night, a background process matches up our messages and tries to optimize some of the links based on connections. So if there are two trees of knowledge there may be optimizations between those two trees. E.g. Knowledge of Computer Science programming and watching how DNA replicates where one could formulate an "Algorithm".

For an Android we would hope to have the background Message optimizers running all the time. Typically it makes more sense to do this when one is sleeping because the days data is in place and no new information is coming in.

OPTIMIZE MESSAGES WHEN IN SLEEP MODE (SC CONTEXT SWITCHED OUT)

Figuring out the closest match

We can use a message based algorithm where we return message sets iteratively and we exclude the messages which have the least message return sets. An algorithm needs to be defined here.

FIND RELATED MESSAGE FOR CAR X

Routing of messages And The Message Bus

We need a mechanism that returns messages from the neurons and passes them to a "Memory Message Bus for the SC". One can imagine that the messages have different weightings and semantic context on this channel and are routes to the SC for real time processing in the Memory message Queue where it is merged with the other Message types.

e.g.

SOUND CAR

IMAGE CAR

CAR

What is semantic content? Message Relationships

Semantic context is the amount of other messages related to this message. For example

CAR

related to many message

AUTOMOBILE

TYRES

WINDOW

LIGHTS

etc;

Therefor this Message is Semantically rich.

A

is related to messages

a

sound a

letter a

This is less Semantically rich.

If we see for example a new language like Egyptian Hieroglyphs all we can store is the message

picture hieroglyph-X

Therefore on its own, it has no meaning and the SC needs to grow Message links to it to provide it with what we call “Meaning” which means the related Messages.

What does it mean to be a type of person? Goal Messages

Using this design pattern the cells in ones body can generate messages in the core part of the brain-stem close to the System Consciousness which generate Goal Messages. These are part of the DNA of the person.

e.g.

GOAL I WANT TO BE AN ELECTRICIAN

GOAL I WANT TO WORK WITH MACHINES

GOAL I LIKE VIDEO GAMES

GOAL I LIKE SPORTS CARS

GOAL I LIKE TRAIN SPOTTING

GOAL I LIKE TO COOK

Typically these need to be triggered.

Therefore these messages also have some meaning. In this design pattern, certain “traits” are passed on as recorded messages. There are also message triggers so that when one sees something like wires on a breadboard, the message is matched and then triggered.

I WANT TO BE AN ELECTRICIAN

and that path is followed and the being specializes in that direction. One often sees people say – When I saw it or read about it, I just knew that was what I wanted to do. In many cases, one then says – one follows in another's footsteps.

Traits and Memories

In this design pattern quite a lot of information is passed in the DNA in the form of stored messages.

Salmon know where to spawn. Video messages explain where to go and when.

Spiders know how to build a web. Video messages are in place to show the different cases.

Birds know when to migrate. Messages containing the rules are passed on.

Once again there are triggers and Goals.

For a Spider the strong Goal is

BUILD A WEB

Human traits

Being good with X. This is also passed on. Therefore the brain passes lessons to the DNA for the next generation.

LESSON HOW TO SWIM AS A BABY

This is later lost as the child grows older.

LESSON HOW TO DO X

Typically in a more intelligent being the Messages are more dynamic and less hard coded but the learning message algebra is more developed.

Message Algebra for understanding Math

Message types that define logic, axioms.

Math Consciousness which has messages which understand what solution to apply

Message Algebra for understanding different Languages

Messages which understand grammar and syntax of a language.

Messages which can map one language word to another.

Message Algebra for Video, Audio, Sight

Messages for sound types

Messages for sight type objects

Run Time Systems for Cell Types

In this theory, it's not enough for a specialized cell to be located in a certain part of the body, it must know the context in which it is running. For example, it may be the middle of the night or other cells around it might have completed some programmed change. Therefore, the cell needs to know what time is it as I've already stated but not only that it needs to know what genes it needs depending upon the context.

Therefore messages must switch on and off certain parts of DNA. Total DNA is the entire program so context goes something like

1. Cell Type
2. Age of Cell
3. Current state of Cell ← current epigenetic markers (see below)
4. Incoming messages from Cell Receptors

Based on

NEW MESSAGE ARRIVAL

1. Pass Message into Context
2. Switch on/off certain parts of the DNA program

At the current time of writing, the name of this state based mechanism is covered under the term Epigenetic (Markers and the like, please search for this for more information). This appears to be the term used for the dynamic messaging to cells and altering states to know its state and build new cell state (but I'm no expert in the area so this is a best guess based on some limited reading).

A simple use case is plane travel where people suffer from Jet Lag. When the time zone changes, it take a body a few days to adjust based on incoming Light/No-Light (day/night) changes due to a different timezone. These signals get turned into

DAYTIME NOW

NIGHTTIME NOW

messages. When the Cells receive these signals, they switch on/off certain processes at the DNA level relating to day time and night time and change the state of the cell to reflect this. Reading up on this the mechanism appears to be the coiling/uncoiling of the DNA in certain sections for on/off.

Jet lag occurs according to this idea because the cells has a switch over time based on the cell timer. It learns this time over a few days.

CELL TIMER SWITCH OVER TIME 12 PM (learned)

Hop on a jet and travel somewhere distant.

NIGHTTIME NOW 6 AM

Jet lag is 6 hours.

CELL TIME ADJUST SWITCH OVER BY ONE HOUR

So it'll take about 5 or six days to adjust unless some treatment is available.

An android can just change locale.

In terms of DNA Apps, how does this work?

In the case of an Android, the way this works is that each Actor will contain the total program. Depending on the context program sub-routines will switch on and off. To use a modern parlance one can imagine all the DNA Programs being contained in a Cell App Store.

The per-Cell App store knows the current DNA Apps installed and running.

Depending on the new Message and its Context, certain applications will be downloaded while others will be stopped and uninstalled.

One can imagine something like

CELL APP STORE RETURN APPS FOR MESSAGE X KNOWING CURRENT STATE Y

INSTALL NEW APPS

REMOVE OLD APPS

So the message probably need to know what to install and what to remove.

This idea works well for day processing versus night time processing.

There are also some discussions about Twins being different based on this environment and this is the same idea where the parent / environment influences the messaging in the child and their growth behavior and preferences changes as a result.

So we can think of any living being as a context driven, message driven being.

Chapter 4 – DNA Apps and Messages

We'll talk about an "App" design where messages are sent between them to control a particular function.

DNA Fragment is an App

Moving forwards, we can think of a switched on DNA segment as an App within an overall App store to use a Computer Analogy.

An Android should therefore have some kind of Android internal App store based on some kind of current state and current message lookup mechanism implemented per cell / Actor.

Bad Cell States and Good Cell States

Using this App analogy we need the right Apps in place **at the right time**. It's important that these are not messed with by outside gene therapies which are faulty and do not understand the protocols at work. This can cause the patient to become ill or unwell. In the future patients who under-go autopsies should ideally have their related cell states checked out if some major failure occurs which has no other obvious cause.

In the case of an android, or system crash, the Actors should also have state which is readable.

Building An Evolving Three-dimensional Life form

How can this be achieved using this design pattern?

Let extend on the current ideas.

First there is a DNA App responsible for building the life form in question. Let's call it

3D LIFEFORM BUILDER (DNA)

It has basic ASSEMBLER message operations like

MOVE
ATTACH
STRETCH
ROTATE
CREATE
REMOVE

and so on. This protocol has both a Virtual and a Physical operation.

By this I mean

MOVE ARM
produces a physical movement of the arm
and an imagined movement of the arm.

The same is true for operations to MOVE CELLS and CREATE NEW ORGAN.

In addition to this, we also add a ENHANCE APP which is a sequence of time based changes for the three-dimensional being. This is analogous to the Darwinian idea but is expressed in a MESSAGE BASED PROTOCOL. These are based on the IMPROVEMENT MESSAGES I discussed earlier which become ASSEMBLER operations like MOVE, STRETCH, GROW and so on. In this design pattern, everything has an ENHANCE APP of improvements to the whole being can evolve (bones, cells, eyes, legs, liver, heart etc;).

Therefore we calculate absolute locations based on the summing of the ENHANCE APP plus knowing the current Cell time T.

So in the DNA there should be something like

DNA ENHANCE APP

which contains the enhancements for everything to build the life form.

Possibly the core DNA 3D Life-form program has an associated ENHANCE APPLICATION

e.g.

GET NEXT 3D ASSEMBLY MESSAGE FOR CELL AT X,Y,Z LOOKUP ASSOCIATED ENHANCE APP KNOWING CELL TIME T

produces message

MOVE CELL TO RIGHT

CREATE A KIDNEY (which creates more Kidney cells from Stem cells)

ATTACH BLOOD VESSELS

So we need a basic BUILDER message API.

A DNA 3D builder App which has an associated ENHANCE APP (These may be one and the same).

A fern has more Chromosomes than a human being (but is much older generically than a human) so this may indicate that the ENHANCE APP data gets Appended to the DNA and is looked up from the “Core” DNA APPS like 3D builder.

So for an Android, we need an Appstore 3D BUILDER App to construct the Android. If we want to give it an AddOn, we add some assembly code to ENHANCE Manager App and this adds the Physical Copy and a Virtual Copy and uses some Core Assembler operations, plus the physical add on in question. E.g. Android gets night vision or New Tool Assembly for some Rig.

Stem Cells and Cell Types

There are approximately 200 different cell types in the human body. According to the design pattern, they have an ENHANCEMENT app which tracks their function based on their context so these summed differences make up a Muscle Cell for example.

A base stem cell can become any type of cell depending on the timing and signaling it acquires. Therefore, it may take “n” messages each arriving at time t to make a Neuron. Also, certain DNA Apps need to be switched on. So how can this be figured out? The answer is to study the life-cycle watching the **signals** and the **timing** and then try to reproduce. An optimization in the human body is to create Base Stem Cells which already have a certain amount of signaling and messages received. From these for example, one could create different type of Muscle for example.

So we have

CREATE STEM CELL ACCEPT ANY MESSAGING, NO STATE, NO DNA SWITCHED ON

Also a base Neuron Stem Cell

CREATE NEURON STEM CELL ACCEPT ANY NEURON MESSAGING, BASE NEURON STATE, BASE DNA NEURON APPS SWITCHED ON

SEND CREATE MEMORY NEURON MESSAGE TO BASE NEURON STEM CELL

So how do this work for an Android?

The idea here is that this is a Computer Science idea of a Base Class Actor (Stem Cell). Other stem cells contains sub-specializations which are sub-classed off the base class. Therefore this idea should be used in an Android design.

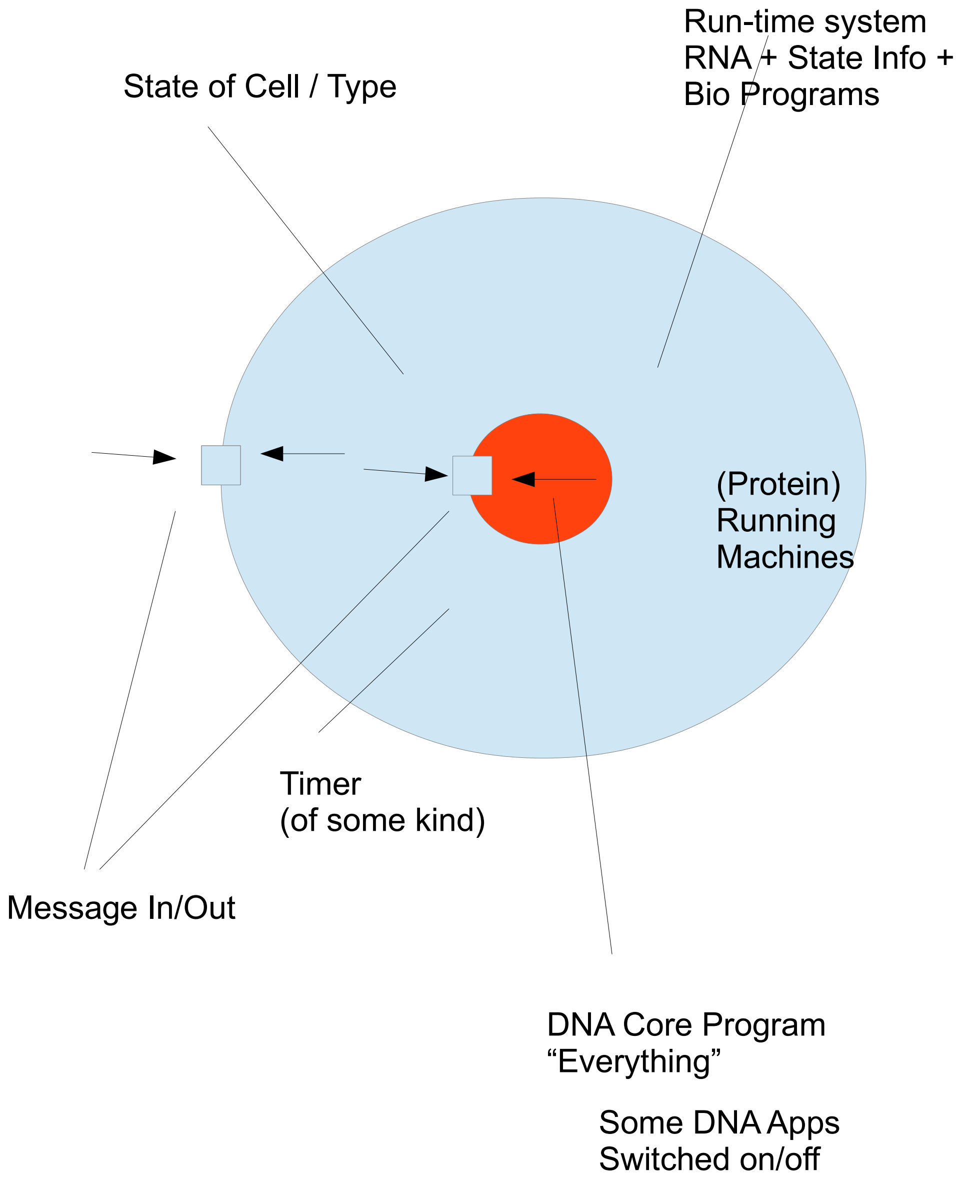
CELL CONTEXT should include

CELL TYPE

CELL BASE TYPE

Cell Runtime versus Stored Program

Here we treat the Cell viewing it in a software way.



DNA and Byte Code

The four chemicals in DNA are a byte code for a bio-program which contains many apps. Some of the apps handle the specialized cell types. Each cells has a run time system which is in the Cytoplasm and a location for the total program which is in the nucleus. RNA is the strip of runnable code and is processed by the equivalent of a protein based Turing machine. The output is specialized protein machines which performed specialized operations. The proteins are also state based machines which send and receive messages also which are encoded in proteins. There are also timer mechanisms for many of the messages.

An Android should ideally support a messaging algebra which can support code generation based on messaging.

e.g.

CODEMSG IF

CODEMSG VAR X < Y

CODEMSG THEN DO SOMETHING

In a living organism these messages are turned into Protein responses. In an android they could possibly sit on top of a Java Virtual Machine for example.

Therefore we can program by messages alone once we have the app.

So we need a CODEMSG App

Attacking The Cell Nucleus

A virus attacks the cell and inserts its code and forces the cell to reproduce it and then destroys the cell. This is a DNA based virus.

This idea here is to prevent untrusted users to rewrite the A.OS core programs without sufficient permissions.

Attacking The Cell with RNA

RNA is the run time code and a virus can contain this. This inserts its code into the run time system of the cell first altering its behavior. Based on some limited research (this is very theoretical and would need to be verified by real lab testing), it appears that HIV may possibly alter the messaging system of a t-cell using this approach and add additional cell messaging handling. Later the RNA virus which has protein machines present zipped inside it converts the run time code to DNA and adds it to the DNA apps and forces it to make a new copy of the virus. This way all the messaging of the t-cells are altered system wide over time. At a certain point in the future, a message is transmitted to shut down the t-cells probably based on some kind of timer but it may be something else. Therefore, the idea is not to destroy the t-cells but rather to reprogram their message handling system wide and then shut them down, rather than have a DNA based virus destroy the invaded cell and then try to explode them so to speak. Possibly one could use the same invasive approach to fix the broken messaging unless the RNA virus was built to defend from an attack like this.

The idea here is to prevent run-time code from being altered while it is running. This might be like a program which sits between the real programs and the inserted one and filters the messages, altering them for its own use.

Orchestrator App, Conductor Messaging and Growing

When a person is growing before reaching adult-hood, one is changing ones structure. This is a complex process. So far I have indicated the need for a 3D Manager App and an Enhancement App. However, there is also a need for another type of App which shall be called the Orchestrator App. The basic idea is that when one is growing, there needs to be synchronization between cells types which are different. So what we need are Conductor Cells. These contain the blue print of the body based on time t. Individual cells can perform individual tasks but timing and operations across cell types is important. The analogy is that of a Conductor of the Symphony. The conductor sends out signals to **“symphony of cell types”** based on the music / instruction sheets and is responsible for the timing of the music / connections amongst other. So we can image a fetus growing and Conductor cells which are short lived (a form of Stem Cell) issuing commands

HYPOTHETICAL SPINE / INTESTINE CONDUCTOR

```
CONCURRENTLY SEND COMMAND
BUILD SPINE
BUUILD INTESTINE
WAIT UNTIL BOTH ARE COMPLETE, HANDLE DONE SIGNALLING FROM INTESTINE, SPINE
ON COMPLETION
ATTACH NERVES FROM SPINE TO INTESTINE
END CONDUCTOR
SPAWN NEW CONDUCTOR FOR OTHER ORGANS
```

The Orchestra DNA App knows which Conductors need to be spawned sequentially at which point in time t.

So we have

ORCHESTRA APP

CONDUCTOR MESSAGING

CONDUCTOR CELLS (STEM CELLS RELATED TO GROWING)

The Cells either transform into the next conductor or they decease and are replaced. These types of Cells are not present in a fully grown adult theoretically. Their numbers should drop off as one reached adult-hood and be larger in someone younger.

An Android should also contain these components during the assembly period.

Regrowing Limbs and Regeneration

Certain aquatic life forms can regrow their limbs. One thing to note about this fact is that the ability to regrow limbs declines with age. Also a human can regrow a finger tip for example but that stops at adult-hood, therefore one can conclude **theoretically** that the regrowth cells are no longer present. As I have already described there are specialized cells for regrowth which I term ORCHESTRA and CONDUCTOR cells. These handle high level component building and timing and wiring issues. In the case of Motor Neuron disease what one needs to do according to this theory is to **identify the CONDUCTOR cells responsible** for wiring up Neurons to the Spine, not just the regeneration of the Cell type in particular. How can we do this? The clumsy way is to try and observe growth. The more intelligent but technically harder way is to reverse engineering the DNA for the ORCHESTRA section of the DNA, so it's probably best to try and identify the Cells involved and find out where in the DNA they are defined.

Theoretical regrowth of limb involves (I realize this is **not easy** to do in the real world!)

PERSON LOSE LIMB A

IDENTIFY CONDUCTOR CELL FOR LIMB A GROWTH

GENERATE CONDUCTOR CELL

INJECT INTO PATIENT IN APPRAPROPRIATE LOCATION

(POSSIBLY THE LIMB COULD BE REGROWN INDEPENDENTLY)

So why do certain life-forms automatically regrow? The answer according to this theory is that the Orchestra Cell types are still in place for growth and the life-forms is still “growing” so a REGROW message is sent whereas in a human they are no longer in place as mammals (possibly) remove Cells like this once they have done their job. Maybe it's a mammalian safety mechanism to prevent multiple limbs from being grown accidentally. There might be some clues in the DNA location here if this occurs in a patient. It could be that the Orchestra cells for this case are still present and sending commands to regrow limbs (theory). Of course any patient must be handled with the due care they deserve.

An Android should also have these components while it is being assembled if it is damaged.

LIMB DAMAGED

REPLACE LIMB

NOTIFY OPERATIONS

Message Optimizer

We need a Message Optimizer App which builds more complex protein machines from similar recurring messages.

E.g.

do A - build AA

do B - build BB

do C - build CC

do A - build AA

do B - build BB

do D - build DD

optimize by building a AB message

do AB - build AABB

Use the approach to path of least messaging (more intelligent messaging). This also means we need more complex builder interface (protein machine).

A good example is a protein machine taking just a single message for the protein type, the protein machine does the mapping based on the complex message type.

In an Android, we need Core libraries which support general messaging.

Onion Model for Message Filtering

In the human brain, the newer pieces are built on top of the older part. Using this approach, we can assume that the incoming messages spread up and out like through various layers of analysis, so that the new parts are where we are “most human”.

From an Android viewpoint we want to create something similar, so for example in vision processing, the furthest out part contains the most semantic context.

It must then be routed back in on the “Message Bus” to the SC.

Antigen and Remembering Attacks

Antigens are stored in the body for viral attacks.

An Android needs some kind of Attack App which remembers the attacks that took place.

This is similar to a standard Virus Attack Software already on Dbs. Ideally the A.OS should be the first line of defense so it should be made as secure as possible.

Bone Marrow Stem Cells

Most of human blood is made inside the Bone marrow. Typically, we have Stem cells in here. This is a repository of Stem cells. Also the skin and different organs have some like the skin.

An android should also keep some Types Actors in a reserve pool in case some of the Actors crash.

Induced pluripotent cell (iPS)

An adult cell can be “rolled back” to that it matches a Stem cell (almost). This means turning off DNA apps and messaging.

First achieved using mouse cells by Shinya Yamanaka of Japan in 2006

In an Android, it should be possible to do this reasonably easily by changing the state and type of the Types Actor. The benefit is a re-usable pool of Actors for re-use in the Android.

Message Bus For The Brain And Message Queue Types

We need to send requests to the various brain locations and get responses. In the human brain there are several message bus channels which deal with various HOS Operations.

General message types by molecule

Dopamine

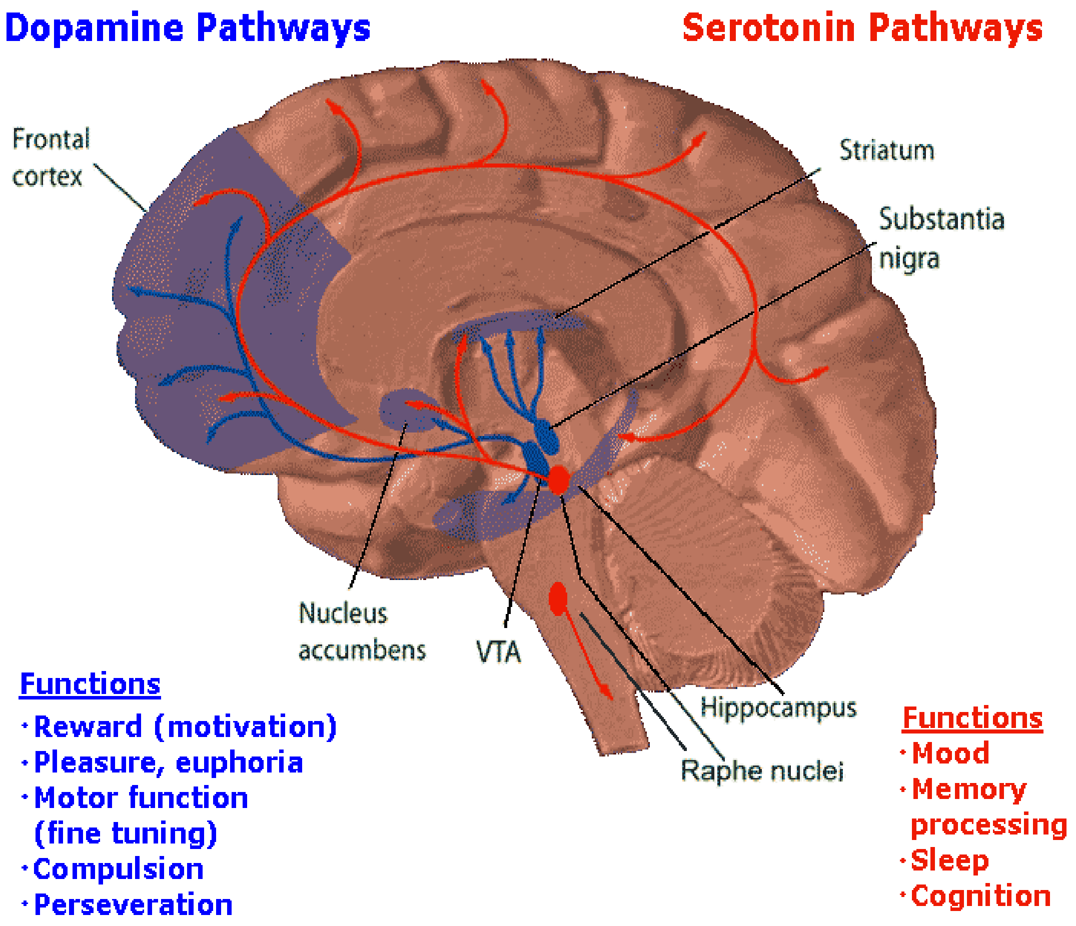
Serotonin

Norepinephrine

Epinephrine

There may be others.

Diagrammatically



The Message bus idea is that these messages end up in the SC.

So using a architecture like this we can imagine that there are several message bus queues plugging into the SC (System Consciousness).

We have (audio,vision, hearing...) as one queue.

Then we have (reward, pleasure, motor function, compulsion...) as another for Dopamine.

For example we might exercise and “feel good” together.

Then we have (mood, memory processing, sleep, cognition).

We wake up feeling “grumpy” and recall the day's work to come.

Why should we separate these out at all? My best guess is that it makes sense that we can both exercise and listen to some music for example **at the same time**. Therefore it makes sense that we should have Message Queue Types feeding into the SC at the same time and they do not interrupt one another. Therefore it makes sense that the have separate Message Queues.

Note: When we say someone is co-ordinated it might mean that they are good at synching these different messaging queues, for example, learning dance steps to music with a partner.

An Android should implement message queues like this.

Chapter 5 – Disease Types And Understanding The Pieces

It is an unfortunate truth that disease unlocks how systems work by presenting a patient exhibiting certain symptoms. Here we cover some problems people have and determine how an Android should work not to suffer these as well.

Prioception And Parkinson's Disease

When people have Parkinson's disease they suffer from Proprioception. What this means is that the Virtual Self is out of sync with the physical self. Therefore the person can imagine doing the task in the Virtual centers of the brain but the physical messaging malfunctions. This for example affects hand movement and a person will rub their fingers together to send data to the Virtual self so that it is re-synced with the Virtual person. So this ties in with the quality of the messaging between the Virtual and the Physical self. One would hope that the Conductor cells can be regrown for the case where the cells are malfunctioning to fix this medical condition. The dopamine messaging is related to this case. It would be interesting to see if it could be possible to create an alternate messaging system from the virtual Self to the hands for example, possibly wireless. I realize this idea is simple to state but would be complex medically.

For an Android, it's therefore important that the messaging between the virtual self and the physical self work correctly and is fast otherwise the Android would experience lags which cause glitches between the virtual and the physical self and would lose its sense of location in virtual space versus physical space.

Language And The Brain

The reason there are so many languages according to this Theory is that human language is our individual attempts to express the Message Algebra built into our human brains with other beings. It's a kind of personalized Algebra mapping mechanism where each group of humans try to figure out some shared rules for sharing the algebra. Interestingly, this Message algebra is also used by other mammals and the like however we have an adaption between brain language centers and our mouth and throat where we can makes sounds for the algebra. It's been shown that the difference between a chimp and a human is mostly in the number of neurons in the brain but that makes all the difference in this case. However, in this theory, the Message Algebra is mostly the same with some additions due to additional Neurons.

So,

MESSAGE ALGEBRA is proportional to the Number of Neurons

MESSAGE ALGEBRA is proportional to the intelligence

However a chimp and a mouse use the same message algebra for moving around in this reality according to this theory.

What we have is the ability to create a LANGUAGE Algebra

Therefore we have

LANGUAGE MESSAGING ALGEBRA

We also have a LANGUAGE DNA APP mapping ALGEBRA to Vocal Chords

The idea is pretty simple

VISUAL ALGEBRA MESSAGE, BOY JUMPS INTO BOAT

SEND MESSAGE TO LANGUAGE ALGEBRA PART OF BRAIN

CONVERT VISUAL ALGEBRA IMAGE OF BOY JUMPING INTO BOAT INTO SOUND

STORE ASSOCIATED SOUND IN MEMORY

SHARE / TEACH SOUND TO OTHERS IN GROUP

The person in question makes some kinds of sounds. According to this theory, the sounds are completely arbitrary **but are based on the underlying MESSAGE ALGEBRA in the brain**. This is analogous to the universal grammar that Chomsky referred to which we all share. In this case, Universal Grammar equated to Human Messaging Algebra which is part of HOS. Some sounds are based on VISUAL MESSAGES or LOGIC MESSAGES or

MATH MESSAGES etc; Some will contain visual actions and so on. From here we create grammar.

So

HOS HUMAN MESSAGING ALGEBRA → (is the basis of) Grammar

Grammar is our attempt to reflect the HOS MESSAGING algebra.

Written Language and mapping to the HOS Algebra

Written language comes less naturally to us than speaking and is a hand-eye co-ordination. The symbolology of human language is also completely arbitrary. However, the key point to note here is that the underlying attempt here is to reflect the HOS Messaging Grammar which every human shares.

HOS Human Message Algebra → Many spoken languages

HOS Human Message Algebra → Many written languages

So what does it mean to learn a language? According to this theory, what this means is that one maps the language either spoken to ones internal HOS Message Algebra Don't forget that the HOS Message Algebra reflects visual situation, logic, strategy, movement and so on.

A simple example is if one says

“Please give me a hand” to a Robot

The joke is that the robot takes off its hand and gives it to the person.

What is really being communicated here is are algebra instructions like

VISUAL ALGEBRA, ROBOT SELF STANDS NEXT TO THE OTHER ISSUING INSTRUCTION

VISUAL ALGEBRA, AWAIT INSTUCTION FROM ISSUER OF INSTRUCTION HOW TO ASSIST

VERBAL ALGEBRA, SPEAK, HOW MAY I ASSIST?

TIMING ALGEBRA, WAIT 'n' seconds

So therefore, **the real task here is to turn all these funny little phrases into the MESSAGING ALGEBRA.**

Therefore an Android needs to be taught what the MESSAGE ALGEBRA is for the phrases of the language in question. An Android therefore must understand a MESSAGE ALGEBRA as complex as a human (maybe even more so over time).

Picking Up A Cup and Co-ordination

It's often stated that a robot picking up a cup is a really hard thing to do. This is true if an Android does not have a virtual self and a physical self. If an Android has a virtual self and a physical self and they are properly synced, then the planned movements are first performed in the Virtual self and then the physical self just does the motor movements. According to this theory, this is how the human brain makes this task "easy" (I use this term advisedly). They key to this is that the Android messaging is performed in milliseconds and the nerve signaling is clean. Otherwise the Android would experience Parkinson's like symptoms.

Context Messaging And SC Actions

Popular experiments on mice and animals show that they can be trained to press levers depending on the Context they are in. Dopamine levels are measured and under certain circumstances a mouse will press a button when they are raised.

We can see the dopamine messages carrying context information e.g. MIDDLE LIGHT ON

Learned behavior

MIDDLE LIGHT ON

PRESS BUTTON

FOOD APPEARS

So when we see a MIDDLE LIGHT the mouse presses the button and is fed.

The context is MIDDLE LIGHT which is provided by the VISUAL MESSAGING SYSTEM. This is passed through the SYSTEM MEMORY and the action PRESS BUTTON is generated based on a MESSAGING MEMORY where the MOUSE WAS FED. Being FED is a goal the mouse has so it's matched and the mouse presses the lever.

The operation

ACTION ALGEBRA, PRESS LEVER is passed into the SC and it causes the mouse to do this operation.

Therefore in this theory, muscle actions are handled by ACTION ALGEBRA.

Certain decisions produce this message however in some cases scientists have found that by stimulating certain regions of the mouse brain so that it keeps doing the action until it is exhausted (not very fair on the mouse). In this theory, the idea is that PUSH LEVER message is fired over and over again and the mouse SC must act on this operation.

Here we are dealing with

MATCHED GOALS e.g. BEING FED

CONTEXT e.g. MIDDLE LIGHT ON

EXPECTED RESPONSE in memory generation ACTION MESSAGE PUSH LEVER

SEND TO SC for Processing

Visual memories are stored as VISUAL ALGEBRA and is part of visual problem solving. So to a certain extent this is a little like a look up where the mouse has message context memories which matched some kind of goal.

Attention Deficit

This is when there is a problem with Message Receivers in the learning centers according to this theory. Messages are not processed as fast as the information which is being received so information is lost. Typically a stimulant is used to speed up the (dopamine) receptors or the information is taught more slowly (personal preference).

An Android should process information as fast as it is being received.

Hyperactivity

Too many ACTION operations are sent to the SC from an over-active goal for interacting with other kids for example.

Short Term and Long Term Memory

Short term information is queued up for possible long term storage.   Need a Message Algebra for the different types.

SHORT TERM STORE PHONE NUMBER

LONG TERM STORE PHONE NUMBER

Note: There should be some kind of weight for information based on how often it is looked up and how many references there are to it.

Mapping A Brain Signal to a Message Type

The Neuron electrical structures and signal strengths indicate the message algebra type.

An Android may need to have some kind of mechanism like this where the Message Types for the Actor neuron has some defined meaning in the SC.

Addiction, Will Power, Goals and Strong Neural Pathways

In this theory the brain is primarily driven by a lot of goals, some are system based and some derived. For example, if we are hungry we need to buy some food and so on. The body rewards good behavior and has messaging to reward us. Our mood and emotions feel good.

EMOTION FEELING GOOD

based on

SYSTEM GOAL HIGH PRIO FEEL GOOD

However some drugs like Nicotine stimulate the production of these messages cheaply. Because the body is goal driven, the need to feel good is strong and pathways are recorded for how this is achieved. The stronger the pathway by repeated behavior, the more we'll do this

GOTO SHOPS
BUY CIGARETTES
SMOKE CIGARETTES
EMOTION FEELING GOOD
GOAL ACHIEVED

The more times this works, the stronger the need to perform this path to satisfy this behavior. It also does not require a lot of work to do (increased local taxes are a way of making it harder). Our goal driven nature to FEEL GOOD is usually achieved by eating a good meal or exercise for example. Therefore our goal driven mind forces us into what we call the ADDICTIVE PATHWAYS or strong behaviors. Addictive personalities are theoretically where the System priority to feel good is set very high. In this case the System Consciousness is getting this goal request a lot.

One often hears talk of 'will power'. In this theory, will power is the desire of the mind to **set its own goals**. This can be a problem when the FEEL GOOD goal has a greater priority than the GIVE UP CIGARETTES goal.

Goals are mapped to Neural pathways in the Memory system and are stored as memories of different weights.

Therefore an Android should have a Goal App and the ability to set GOALS for the Android and have different priorities which determine how often SC received this goal request. Typically the body knows best but in the case where the PRIO causes a problem for the Android, it should be possible to over-ride it. We call this STRONG WILL POWER if one can maintain a new goal over a system goal. A lookup mechanism should find high priority behavior stories to achieve goals.

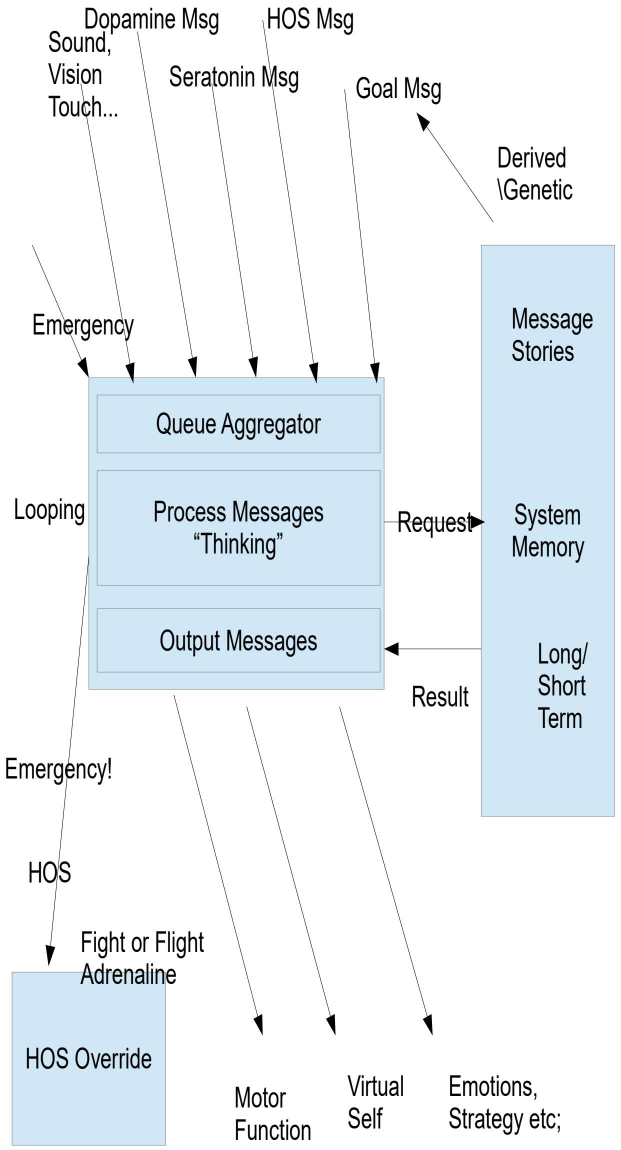
So we have

GOALS of differing priorities messaging SC

PATHWAYS of differing strengths to achieve goal (most strong path first based on feasibility)

System Consciousness Architecture Diagram

Let's bring the ideas together discussed so far into a simplified diagram. This is just the System Consciousness. We only break out to the HOS where there is an emergency e.g. chased by tiger.



Next I'll discuss what I mean by a “Message Story”.

Memories And Message Stories

When a message is looked up in the system memory, one or more “stories” can be returned each with differing importance and appropriateness with the search terms.

The idea of a message story is that it contains not just a list of Message Algebra types but also the **sequence** of algebra types.

A simple example is consider the search words **fixing a wheel on a car**

If this had already been done before and memorized, then a list of Algebra Messages are retrieved.

VISUAL ALGEBRA PUT ON HAND BRAKE

VISUAL ALGEBRA FETCH CAR JACK

and so on...

the next best result might be

VISUAL ALGEBRA FIND BOOKLET

Therefore the time based steps are also present in the sequence of tasks.

These can be thought of as stored memories using normal parlance.

In this theory, some remembered memories are present when a child is born commonly referred to as reflexes amongst others. In this theory, a spider is born with a goal for its new home.

GOAL HIGH PRIO BUILD WEB

MEMORY STORY, BUILD WEB, START

…. some Message Algebra instructions

MEMORY STORY, BUILD WEB, END

Acquisition of Language

Studies have shown that language is acquired while a child is young and if this windows is missed is it very difficult to learn languages. In this theory, this period of time is when the brain is in learning mode and a LEARN_SPEECH App is switched on in the brain. Most creatures have this according to this theory but because the human brain has more Neurons, the SPEECH Algebra has a wider range of sounds. The basic operation of the LEARN_SPEECH App is to map the sounds to to the Language Algebra. After a while it's turned off. Therefore if a child never learns a language like a feral child for example, it still has the language algebra in its mind for inner thoughts so it can make decisions about gathering food and so on but has no capacity to make that thought algebra into sounds.

Socialization of A Growing Being

In this theory which runs contrary to Darwin here and there one of the ideas is to do with how the bodies' inner cell messaging switches on and off certain growth phases in the brain and body based on the quality of the caring the young life receives. Therefore positive parental stimulation produces both physical and emotional growth. It's been shown in some feral cases of children and even in poorly run institutions that a lack of interaction with other humans which are not caring or present causes many serious issues in brain development and physical development. It's also been shown that monkeys without a loving parent grow up damaged. Why is this?

The rough answer according to this theory is that for one being to care for another as an adult, it must first be taught love and caring as a child. The mechanism of reality appears to stunt the development of non-socialized primates for example that do not receive this care in order to promote socialized behavior where one looks out for each other and may be beneficial to the overall group dynamic in the longer term. This logic is **purely theoretical** and may not be entirely correct and studies should prove this to be true or false.

However, if one creates an Android it makes sense to me anyway that it must learn what it is to be cared for at an early stage of its development and should not develop if it does not receive this. This may seem optional but nature does seem to do this for us and (I guess) this is the reason why; that machines will be safer to be around and watching out for you if they know what it means to “care and be cared for”. I don't have conclusive proof for this idea but it springs from the idea of some people commenting on other parents as “good parents” when they spend time with their children and hug them and interact with them in a caring way. It appears that cells message each other in a good way when this happens and become better “well adjusted” adults in theory both physically and emotionally. Like I said, I don't have conclusive proof for this idea and I am sure that there are many exceptions to the rule but I think this should be part of “Android Development”.

Improving Language Translation

It is notoriously difficult to translate from one language to another using a computer and there are numerous sometimes humorous examples of how machines get this wrong. Why is this? According to this theory, the reason for this is that the sentences may have several underlying MESSAGE ALGEBRA equivalents. Recall the sentence “Give me a hand”

English

give me a hand

German

geben Sie mir eine Hand (this may or may not work depending on whether German uses this idea)

Very quickly this can go wrong

If we translate to Punjabi

ਤੁਹਾਨੂੰ ਮੈਨੂੰ ਇੱਕ ਹੱਥ ਦੇ ਸਕਦਾ ਹੈ

this then translated back to English as

You can give me a hand (which turns it into a more “literal” meaning)

What is really being asked here is

Can you help me please with this task

which is

können Sie mir helfen mit dieser Aufgabe bitte

Also Babelfish translates this as

Du kannst mir mal helfen

and Google Translate as

geben Sie mir eine Hand

Now, if we had first converted this sentence in English into Message Algebra as the base type, then we would have a better base language from which to convert to another language

source language → language message algebra → target language

According to this theory, this is why we as humans produce better language translations because we are using the Message Algebra in our brains before mapping from one language to another. There may be a visual context for example which the words in and of themselves do not describe and the MESSAGE ALGEBRA does. This would require some technical work by language experts on creating a deep semantic rich LANGUAGE ALBEBRA but should improve the quality of the translations longer term. I am not sure if this would totally remove the need for language translators but this approach could also be useful for completely unknown / dead languages also.

An Android should therefore have a good base LANGUAGE ALGEBRA for a language and use this mapping mechanism to other languages. Basically, the Android needs to support this intermediary form and pluggable language mappings (probably provided by the speakers of the language.) Therefore they turn the LANGUAGE ALGRBRA into their language instead of going from another language like English for example in theory.

Reading A Text And Answering A Question on It

By using this approach, if we turn anything we read into this Message Algebra as Story Message Blocks (what we call Chapters for example) we can Query the Message Algebra which is stored in Message Blocks (Stories).

Therefore we can deduce things from the Message Algebra

VISUAL ALGEBRA ROBBIE THE ROBOT MOVES BESIDE ASTRONAUT
VISUAL ALGEBRA ROBBIE GIVES A SPANNER TO ASTRONAUT

Question

What did the robot give the Astronaut?

QUERY ALGEBRA What did the robot give the Astronaut

ANSWER:

RESULT ALGEBRA A SPANNER

SPEAK RESULT TO LANGUAGE

“The robot gave the astronaut a Spanner”

We therefore need an ALGEBRA for this situation as well.

Gamma and Glutamate and Controlling Signal Transmission

Gamma and Glutamate lower and raise signals traveling through Neurons. Alcohol is known to lower the signaling and lessen thinking overall for example. This allows Neural pathways to be controlled in this way. It makes sense for an Android architecture to lower the signals passing through Neurons or raise them. This is part of the Actor Cell Type configuration. This can be used to fine tune Neural pathways.

Facial Expressions and Emotional Algebra

Facial expressions need no real explanation except that they are a manifestation of Emotional Algebra mapped to facial muscles.

An Android should support this where possible.

Dreaming In Another Language

One dreams in another language when the Neural pathways are stronger in the language centers and LANGUAGE MAPPINGS MESSAGES are generated and sent to the SC.

Context Switched Rooms In The Mind

Sleep is a bit of a mystery for people. Often dreams can be very realistic but when one wakes up, one may not be able to remember the dreams and if one does, one may not understand the meaning.

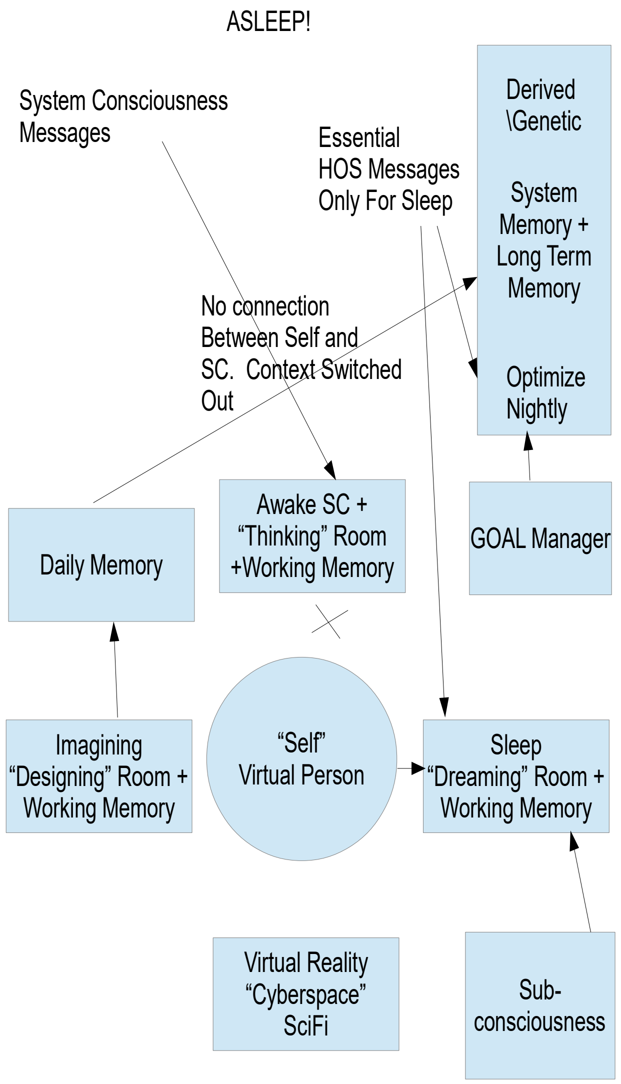
In this theory when one sleeps, one switches to another room in the mind and the Self is placed in a “Dream Room”. The Self is a virtual copy of ones body which normally syncs with the real world “room”. Messaging from the body is limited in this dream room and one goes into a maintenance mode where the mind organizes data from the day that has already been passed in. Limited HOS Messages are sent into this room and one sleeps here. However one can hear or be shook awake. The idea for this mode according to this theory is to allow the body to do work in the Conscious messaging system and allow the Self to “disconnect” or sleep.

Typically sleep allows one to “gather one's thoughts”.

Also, while we are awake, it's possible to think inside another room where one can work on a problem and context switch between the rooms dynamically.

Also, when one sleeps, the HOS sends GO TO SLEEP messages to the CS are based on Cell timers.

While we are waking up we may be in both rooms temporarily and have some idea of our dreams shared with SC which connects with the “awake” state.


ASLEEP!
System Consciousness Messages
Essential HOS Messages Only For Sleep
Derived \Genetic
System Memory + Long Term Memory
Optimize Nightly
No connection Between Self and SC. Context Switched Out
Daily Memory
Awake SC + “Thinking” Room +Working Memory
GOAL Manager
Imagining “Designing” Room + Working Memory
“Self” Virtual Person
Sleep “Dreaming” Room + Working Memory
Virtual Reality “Cyberspace” SciFi
Sub-consciousness

An Android should support “Virtual Rooms” based on Context Switching like this. Sleep can be seen as a Maintenance Mode for the Android.

Building A Room From Visual Algebra

The room that the Self inhabits is based on the more complex messaging protocols. It is the equivalent of Open GL in software terms. The messages that are received from the Auditory, Visual and Central nervous system define the contents and the feel of the room for the Self.

The image processing centers build object primitives and then apply texturing.

VISUAL ALGEBRA CREATE CHAIR

VISUAL ALGEBRA CREATE ROOM

and so on. This is a rich language with many descriptions of the objects in question. This is the language which builds the rooms that we inhabit in real time. The VISUAL ALGEBRA is based on image analysis and its best guess. I've talked about their earlier.

All the objects and things in the room have built in “semantic context”. Therefore if the Self is looking at the objects, the person automatically knows via messaging what the object is.

This is a fine grained process where each part of the place in question is broken down into semantic units.

Therefore we tag all the vision actors with semantic context e.g. each dot in the circle sign “edge, sign, red, circle”. Semantic info is sent down the message bus into the SC and store in virtual memory. For example a fly just has limited messaging “circle, landing surface”

Major Types of Semantic Messaging

Here are a high level list of the type of Algebra we need.

Reward

Pleasure

Goal

Euphoria

Motor Function

Compulsion

Preservation

Mood / Emotion

Memory Processing

Sleep

Cognition

Visual

Audio

Vocal

Math

Logic

Reason

Language

Strategy

HOS Events

Using Cells to implement a Turing Machine

This shall be achieve using the messaging algebra.

How to make an android feel pleasure/pain.

Pleasure and pain are manufactured based on a message sent into the SC.

Sex – reproductive messaging

Messaging is sent to the reproductive messaging sites.

Visualization or Yoga

This is when the Self switched into another “Meditation room” in the Mind where it creates its own room based on internally generated messaging.

Savant

This is where a person has more specialized cell types than another in the brain.

Psychic - external messaging mechanism

If this exists, then it is external messaging.

What is in mothers milk? messaging... colostrum

Messages are passed from mother to child to boost immunity.

DNA Protein Machines

Proteins are programs built from code which handle certain messaging

We need protein machines to do - power generators, transporters, builders, guardians, etc;

Hallucinations

Incorrect messaging sent to the SC.

Delusions

More incorrect messaging sent to the SC.

Mental Illness

Problem with the message bus / too many or too few of a certain message type.

Self Control

Controlling message handling via the SC.

Stress

Unable to obtain goal so body has a negative message result.

Relaxed

All current goals reached

Conversation / Writing

Words and sounds are converted into Internal HOS grammar.

Extremely Accurate Memory

Some people have incredibly good memory. They report being able to smell, feel and touch what they remember. The way this works according to this theory is that they store the video message algebra sequentially in real time. Most people don't do this. It's also searchable and they can be context switched to the Memory Room based on facts, similarly to a Creative room for designing. Therefore they have a form of Video Algebra Memory which dynamically builds memory rooms. They can then place themselves in the room and it's completely real to them, then context switch back.

Simple Mechanism for Timing Things And Wait Operations

So we need a timing mechanism based on a state machine which switches apps on and off. However we need a mechanism for timing.

State A; Wait; Wait; Wait; Wait; (4)

State B; Wait; (1)

State C; Wait; Wait; Wait; Wait;Wait; Wait; Wait; Wait; (8)

Therefore we need a Wait mechanism, maybe where each Wait represents a period of time. An Android should therefore have this mechanism.

In the case where living forms have an accelerated aging process, one can think of this as the Wait mechanism is damaged or missing. Therefore the being moves through its various states more rapidly than another person. Various development apps switch on too soon and causes developmental issues.

How can this Wait state be achieved in a Cell? In theory, we can have a stack of specialized “Wait” messages feeding into the Cell nucleus, stopping it from moving to the next state too soon. There may be more than one type. In the case of a human being this could theoretically be a stack of specialized proteins which are attached to the Cell rim and grow smaller over time as they send a Wait protein message to the Cell's Nucleus. Note: **This is an entirely theoretical idea and may or may not be how it actually happens** but the basic idea of a Wait message is one which an Android should use.

Autism

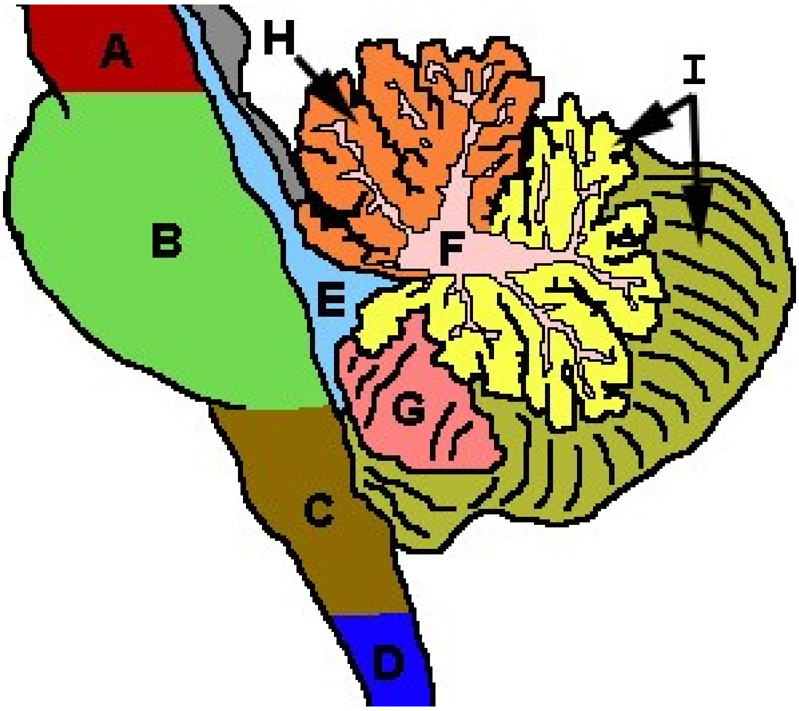
This is a very broad topic. There are many symptoms. For example some children experience a lag between sound and vision Therefore children there tune out from sound and try to rely mostly on VISUAL MESSAGING.

However, let's discuss the core architectural problem here as it's seen according to this theory.

The SC aggregates messaging as I have shown before in the Cerebellum.

Let's dig into this section of the brain in a little bit more detail.

Cerebellum and surrounding regions; sagittal view of one hemisphere.



A: Midbrain.

B: Pons.

C: Medulla.

D: Spinal cord.

E: Fourth ventricle.

F: Arbor vitae.

G: Tonsil.

H: Anterior lobe.

I: Posterior lobe.

According to the theory, all of the messaging converges here into what I call the SC. Now, one important case for mapping autism is where the Messaging is fine but the SC is not functioning correctly so the messaging is either broken or garbled.

Therefore imagine an SC where only certain messaging arrives and other messages do not make it in there correctly. This is due to this part of the brain not properly forming for some reason. It's been shown by cross examining tissue in the brain of some Autistic people that the Cerebellum cells are immature for some reason compared to a healthy child / adult.

Therefore a possible medical solution according to this theory is to inject conductor cells into this region of the brain to "mature" these cells which have not fully developed for some reason. **I realize this is not so easy to do in the real world but this is the theory.**

Why could this happen? Special attention should be paid to any medical intervention to a child which targets this region of the brain and alters its cell messaging at critical periods in its development. For example, Hepatitis can strike this region of the brain and any vaccine should be checked for its ability to interfere with Cell messaging. Recall from the theory, that the **timing** is also important to determine which protein message are handled and how. In some rare cases, it might be theoretically possible to mess with the cell development here and cause Autism like symptoms. **This is very theoretical and I realize that I am not medical and vaccines save millions of lives per year and I am not advocating a lack of use of vaccines! They are fundamentally good.**

Lets try and understand the effect of Autism on a child / person next when the SC malfunctions.

Therefore an Android Critical component is the SC.

Autism and Frustration / Specialized Needs

People who have autism report that they feel very stressed a lot. How can this be explained according to this theory? In the SC, goals are fed into it continuously, for example "Make a friend”. A person will them try to do this and the autism makes this difficult if not impossible. When a goal is not met in the SC, the SC produces Stress Messages which make the person feel unhappy because the goal is not reached. The autistic person has no control over the stress messages; the body is doing this. Therefore it can be really hard for an Autistic person so the best way around this is to help them achieve their goals in whatever way is possible. This also leads to an individual with very specialized personalities focused on niche goals. Over time after many retries of certain goals the person remains stressed but will move to goals which are achievable and may not be people based if the messaging for this is not available due to a damaged SC. This varies from person to person but they will be rewarded by the SC for achieving these goals to they will focus on doing a task which makes them feel good. Therefore one can see autism as forcing a person into specialized interests which basically make them feel good. The goal of course is to come up with some treatment to fix the message blocks if at all possible.

It's been reported that pressure eases stress messages. This may be because it dampens the stress messages in the brain by over-riding them with Skin Pressure Messages. Maybe they have a higher priority as they might be related to an early warning danger system (guess). External stimuli may have higher priority over internal goals. Therefore to prove/disprove this one would need to monitor brain activity when pressure is applied / not applied and then one might be able to locate the precise location of the stress generator and somehow control it.

An Android's interests can therefore be controlled by their goals. Achieving goals makes the android feel good and vice versa if not. It's hard to know if we want to make an Android “stressed”. This is more like a goal retry mechanism.

Competing Messages And Goals

If we adopt an architecture where there are message types and goals all going into the SC at the same time there is the chance that one or more goals may compete with one another. For example

HUNGRY SO MAKE A SANDWICH

vs

TIGER APPROACHING RUN!

Obviously the answer is to run. However in an earlier entry I described the nature of skin pressure and how it appears to lessen stress. This is probably why people when they are friendly with one another give each other a hug. The basic idea is that certain messages types have a higher priority and therefore negate the other message temporarily.

So a simple hierarchy is

(MOST IMPRORTANT TO LEAST)

Adrenaline Messaging e.g. Chased by tiger so Run!

Internal Alerts Messaging e.g. Severe Pain / Sickness etc;

External Stimuli e.g. Skin Messaging, Temperature Change

Goals e.g. Do Something New

General Thinking e.g. Problem solving. Food choice in restaurant menu.

The benefit of categorizing the messages for example with Stress is that if we can figure out what priority it has, we can then possibly over-ride it with a less unpleasant message.

How is this architected in the SC? In the SC different Prio Messages have attached Prios. Biologically I think they maybe on separate circuitry but may not.

Therefore an Android should adopt said architecture where we have higher and lower prio messaging. It's up to the architect of the Android to work this out but above are general guidelines.

Acupuncture

If we adopt the idea that the brain has an SC and it is processing different messages in parallel and that certain messages compete with one another and there is a hierarchy, then in theory all we need to do to reduce pain messages, for example, is to produce a message which overrides a pain message. According to this theory, this is what acupuncture is all about. A technician inserts a needle into the skin of the patient and moves the needle. The patient then feels some kind of tingling feeling and this provides them with some pain relief from some other conditions such as Arthritis. Studies have show that sections of the brain are deactivated by the Acupuncture. The theory behind this is that the body has been designed in such a way (I am not so much concerned by who or what here) that different points in the body when disturbed by Acupuncture generates different message types which can override certain symptoms of certain medical conditions and activate/deactivate parts of the brain. In traditional western medicine one targets the pain receptors directly and using this mechanism one permits the message but overrides it. The key to this approach is to figure out the mappings plus it's cheap and minimally invasive. Studies have show this works in some cases. However, according to this theory this technique it's not entirely clear that overriding the message in question will heal the underlying condition but rather improve the quality of life of the patient in some cases. Later I will cover different use cases of Acupuncture in more detail.

An android should support such a mechanism in case the SC is malfunctioning and needs to be fine tuned. One can think of this as “fine tuning” an Android.

Chapter 6 – Achieving Message Balance

A body needs to be fine tuned or balanced. Here we discuss how this relates to ancient medicine.

Balancing the SC Messages Qi / Chi

In Chinese medicine there is the concept of the Qi which is the body energy. The idea is to keep this in balance. In this theory, we imagine an SC which receives many types of Message Algebra and the idea here is that the messages are "balanced". By this I mean, that not too many of any one type of message is prevalent. This can lead to various disorders. So a healthy body has a healthy mix of messages from all parts of the body.

To achieve this in Chinese culture, many people do exercises to focus their energies. In this theory, this maps to the task of visualization where one imagines the movements and attempts to have the body reproduce them. Simple as this sounds, this is a form of mind body training where the SC becomes more focused and one can visualize certain parts of the body in great detail and focus messages at that point.

From this we enter the realm of martial arts where masters of their craft direct messages from their visualization centers which generate messages in the SC and force is directed to a limb or movements with great speed and accuracy. According to this theory, the circulation of this energy is a send / receive message cycle which produces a healthy body mind harmony. The mind in this case is the SC which is coupled with the Visualization centers.

To a certain extent this is a little like fine tuning a car where we get for example the right mixture of engine components to make the engine run "smoothly". In this case, a healthy mind has a healthy mix of messages and the Visualization center is accurately calibrated with the physical form.

In Chinese culture this is Taoism and tao where we have two opposing components. One can map them to to Physical and the Virtual form of the being which must be in harmony to work correctly and prevent illness. This is in opposition to Western Medicine where one focuses on a specific ailment and fixes that.

In this theory one can think of this as fine tuning the Message Algebra. It makes complete sense.

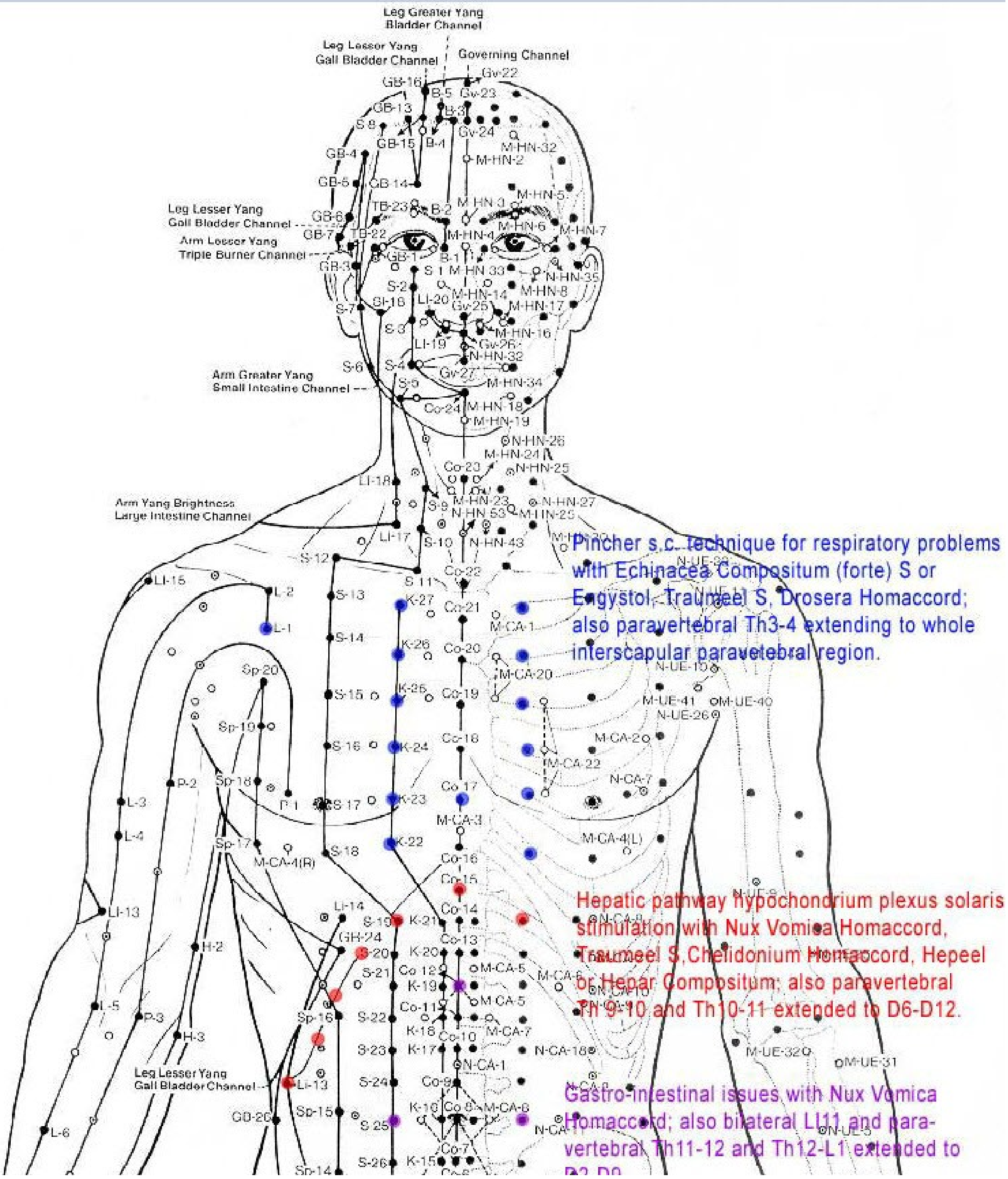
An Android should therefore have a balancing mechanism for the messages and some kind of training routines where the virtual and the physical self are tested just like Tai Chi.

Acupuncture Points

This theory asserts that the Acupuncture points which are present in the body have been designed by the DNA. Their specific locations have been figured out in Chinese culture by trial and error. The idea of the locations is that they generate specific internal body messaging which can override the messages of someone with a particular ailment.

One could imagine some kind of future device where a patients sits on some kind of device which transmits targeted electrical impulses to Acupuncture points without the need to insert needles (theoretical) and would provide pain relief and so on, possibly by massage or near skin contact. This idea is of course very theoretical but I suspect this may have been the design intent in the DNA to help ill beings who shares these pressure points (theoretical guess on my part). One would hope that a non invasive device could be developed. In some videos, one sees Chinese medical practitioners who move their hands around patients. Possibly an Android could perform some kind of procedure like this if they can generate the correct electrical impulses and target them to the body regions needing them.

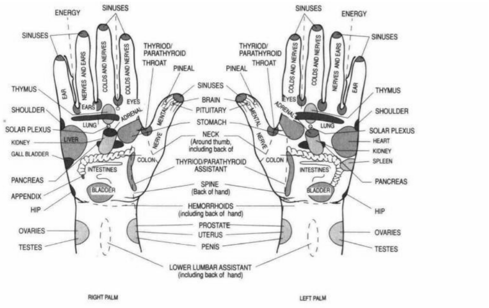
Here are some samples. There are many acupuncture points.



Pincher s.c. technique for respiratory problems with Echinacea Compositum (forte) S or Engystol, Traumeel S, Drosera Homaccord; also paravertebral Th3-4 extending to whole interscapular paravetebral region.

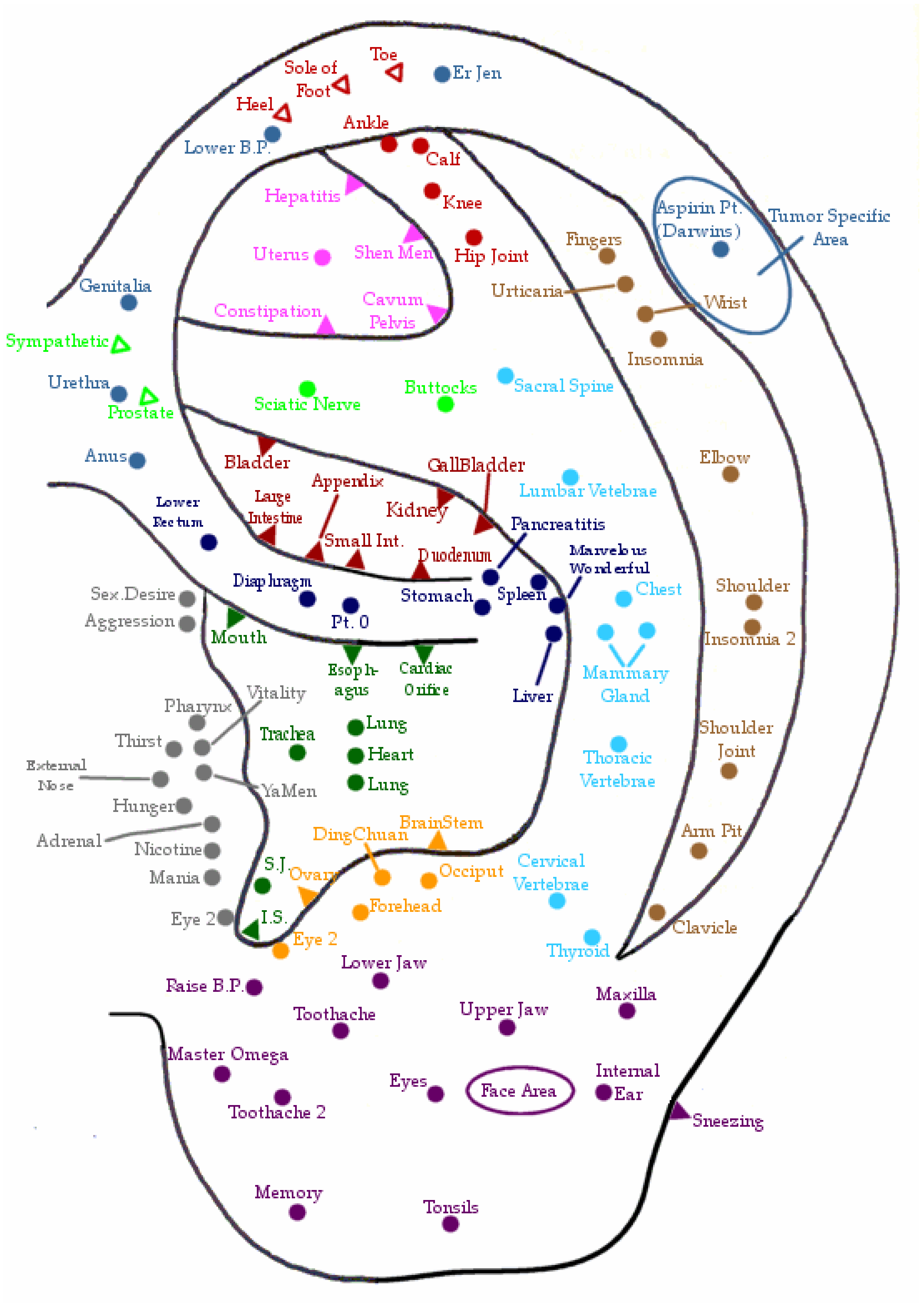
Hepatic pathway hypochondrium plexus solaris stimulation with Nux Vomica Homaccord, Traumeel S,Chelidonium Homaccord, Hepeel or Hepar Compositum; also paravertebral Th 9-10 and Th10-11 extended to D6-D12.

Gastro-intestinal issues with Nux Vomica Homaccord; also bilateral LI11 and para-vertebral Th11-12 and Th12-L1 extended to D2-D9



RIGHT PALM

LEFT PALM



An Android should have some interface points as well for fine tuning.

Alzheimers

This one is a bit of a mystery as to the root human cause so this is a best guess. However it affects a lot of people in middle age and later. The theory is that the Neurons are producing an excess of a protein which in turn affects rapid aging of the Neurons in question. Therefore, one must remain active or do something mentally stimulating to flush out the over produced protein it's not too easy to do this.

Recall before the discussion of the rapid aging mechanism where there were not enough Wait Protein Messages so the aging happened too fast. In this case, the presence of these other Proteins increases the Neuron Clock Tick / aging. In the case of some children with Down Syndrome they have two Chromosome 21 and therefore twice the chance of over-production of the protein. Therefore it's a premature aging of the Neurons according to this theory based on incorrect protein signals due to genetic factors which are switched on. The protein appears to speed up the aging of the Neurons theoretically.

So theoretically one should try to figure out the epigenetic neuron markers before and after the advent of the disease and try to switch them back off again to make the neurons healthier again. Admittedly there are billions of Neurons but the illness cascades from the memory centers. This is my best guess on this one. It's a Neuron timer issue.

In terms of an Android, this is related to Cell timing. It makes sense that they should probably have some form of duplication in case some are lost. This is analogous to people who are good at crosswords and have different Neuron copies in their minds so losing some to Alzheimer's means some backup copies are available. An Android should have one or more clusters of Neurons which are copies. In Software terms, this is called Clustering.

Synesthesia

In a regular human brain only certain message types are paired. This is how we figure out metaphors and the like.

BITTER WIND

TASTE MESSAGING + SKIN MESSAGING

However this is a more general combining of information and the brain does not require it.

The clear essential use of message combining/pairing in the SC is

PAIR (VISION + SOUND)

This is pretty fundamental and key to survival.

This is an example I have used earlier where one combines multiple clicks with light and we get different visual results when they are combined/paired in the SC.

However, some brains do not have strict filtering of MESSAGE ALGEBRA PAIRING it seems.

For example, many people with synesthesia have an SC which pairs

PAIR LETTERS messaging with COLORS messaging
PAIR SOUNDS messaging with COLORS messaging
PAIR WORDS messaging with TASTE messaging
and so on.

We break the MESSAGING into Message Types so IN SC PAIR (TYPE A, TYPE B).

During the learning phase of the human development, these pairings are established.

Also Creative Visual Messaging can combine with Visual messaging producing number lines. These are message that normally do not appear in the regular Visual Window.

So if the pairing does not impair the person, then it's passed on, looks like.

Pairings you don't want are

See Tiger Message and Background flashes so you cannot run.

Mostly these message pairing which work make people's lives more interesting. Sound plus Color is a popular one.

Some pairings have a visual aspect e.g. flashing color, where they enter the Vision Room others do not.

An Android should have some kind of Message Pairing Configuration. The minimum is SOUND + VISION messages.

Allowing features like Number Lines in the Reality Room from the Creative areas is useful but it is probably something that the Android itself would want to turn on/off depending on the situation.

This should be configurable.

Migraine and SC Blockages

In this theory the SC is modeled as a Toll Booth with many stations. There are different channels for different message types of varying priorities. In a healthy brain the MESSAGE ALGEBRA processes the messages as fast as they are received. Therefore there is no problem here.

In the case of migraine, the messages are over-produced in certain regions of the brain by the Neurons and cause glitching or Message Traffic Jams in the SC. The consequence of this is that the Consciousness is flooded with sensation and overwhelmed. The reason for this is that the brain regions must re-balance after some kind of trauma due to stress or alcohol or whatever and may over-produce Messages which flood the SC. When the SC is flooded, goals are not reached and motor functions and visual processing messaging is affected therefore one can experience “glitching” in the Visual Room we live in and holes can appear in healthy vision. This is due to the VISUAL MESSAGE ALGEBRA messing up the Visual Room it is building.

Typically one needs to lie in a room in darkness or take medication of some kind or treatment to lessen those messages and unblock the SC.

It's important that an Android SC does not block up so it goes back to the Chinese Qi or keeping the SC balanced. Typically people are told to eat regularly and sleep properly which is a Qi recommendation. Acupuncture can also work.

However “migrainers” suffer from an SC which can block up easily and whose Neuron centers are prone to over-producing too much Message Algebra. When the Migraine spreads, this is the messaging backing up into the other brain centers.

An Android should have active monitoring of its SC message queues and have alerting for any problems so that glitching does not occur.

Eat vs Don't Eat Messaging and Weight Control

In the developed countries weight control is a serious public health issue. Many folks assume that weight control is a will power issue. The correct message of course is to eat less and exercise more. This makes total sense and is a good message.

However, there are still folks who struggle with weight gain while others complain that they have difficulty regaining weight once it is lost. How is weight control described according to this theory? It's reasonably simple to describe it according to this theory.

First off the SC receives two kinds of elementary messages.

EAT MESSAGE

DON'T EAT MESSAGE

Do not eat suppresses the other message. They types of foods one chooses to eat are looked up based on ones' experience. The nutritional value of this food is independent of the message itself to eat. The desire is to find something to eat.

First point to make is, that this Message is a Priority One Message. This means it's a survival message. It's very hard to over-ride this message but it can be done let's say if one is on a diet but this is why people say it's hard to diet. One is trying to override a priority one message.

The "loudness" of the message is the **frequency** of the message (aka the degree of hunger), a bit like when one gets a Migraine where one is overwhelmed by the messages. If one is born with a message that is "loud" one will find it hard to not eat while others will find it easy to stop and they will naturally say. "You need to have more will power!" According to this theory, their EAT message is not so loud and they have to understand this. To a certain sense, we are biologically hijacked by a message which is keeping us alive. It makes sense that it's hard to ignore this message. The body is trying to keep you alive.

The next point to make is about how some folks suffer from massive weight gain. As I have said already there is a battle going on in the brain of EAT vs DON'T EAT. Biologically we can tell ourselves not to eat as I've said already. One then asks the question: how does the body know its ideal weight? According to this theory, there is a genetically determined WEIGHT RANGE which is determined by one's ancestors based on their experiences.

UPPER WEIGHT RANGE definition

LOWER WEIGHT RANGE definition

Therefore according to this theory, the body calculates the PERSON WEIGHT based on a variety of messages like

ENERGY INSULIN LEVEL messaging

FAT LEPTIN LEVEL messaging

etc;

and so on to determine the present WEIGHT. In this theory, this is a complex calculation that the brain does on our behalf. If one is growing, the body may demand food or issue a don't eat during another period. If one is within the UPPER WEIGHT RANGE then the loudness of the EAT message falls off and one does not feel hungry so much.

As I've described already, some people are born with "addictive personalities" which means that their messaging is loud for one or more conditions. For example, people who really like to eat and enjoy it would have loud/high frequency EAT messaging. These are the kind of people who will talk a lot about food because they are thinking about it a lot due to the messaging. This is perfectly healthy.

However if this is coupled with a high UPPER WEIGHT RANGE definition defined in the genes then the may be in for a life struggling with weight gain and yo-yo dieting. (I include myself in this category). How can one determine one's upper weight range? The simplest way apart from DNA analysis is to look at ones family and relatives in my opinion. This may or may not work in every case.

So how can this be fixed apart from dieting? This theory does not say dieting is bad. This is how one moves up and down within a weight range but one will tend to move towards an "average" weight based on messaging and food preferences. The simplest analogy is that if one likes fast food one will move up towards the UPPER WEIGHT RANGE and vice versa.

I personally think the world is a better place because we all do not look exactly alike however I do get the important point that life-span is shortened by excess weight so it needs to be dealt with. Therefore, the solution according to this theory is to alter the UPPER WEIGHT RANGE and (maybe) lower the lower weight range. I do not say this casually, this needs to be studied by the relevant authorities so another problem is not introduced like life threatening stick thinness.

Obviously if one has no food, one falls below the LOWER WEIGHT RANGE but ones' mind will be flooded with EAT messaging and the body will consume itself until it perishes. Hopefully this does not happen.

So how do we cover the case of a person who was obese and then transformed their weight and kept it off? The answer according to this theory is that they are still within their LOWER WEIGHT RANGE and they have strong/loud messaging for some kind of activity which may be sports related. Competitive sports messaging have a High Priority as these are part of key life skills. This is why we all love a big game in my opinion and why playing competitive sports may help keep weight in the lower range. Typically according to this theory, a person will stay in or around this lower weight range.

The weight range is theoretically used by the brain but the settings are in the DNA.

Do I have any hard proof for this idea? It's been shown that certain medications for brain conditions cause kids to suddenly gain weight as a side-effect. Also they can develop Type two diabetes which is also related to weight measurement. The dopamine receptors are targeted and their messages lessened. In theory here, the DON'T eat messaging and weight information is suppressed and the the brains default message is EAT. This is a bit like a plane where its fuel sensor is not working. In the case of doubt, keep giving fuel to the motor because we know a flight is taking place. So EAT is a default message if no other related messages are received. Also a person who is growing still has not reached their UPPER WEIGHT so they may end up eating like an adult.

Insulin maintains our energy levels so it could be that poor messaging causes this mechanism to break/over produce and stop causing Type two diabetes. Also, this messaging is transmitted in the blood and is turned into messaging in the Hypothalamus.

There should be some Epigenetic markers for Upper and Lower weight range in the Cells according to this theory. This needs to be proved/disproved by experimentation. Any changes should be done **very carefully!**

The conditions which fall under this are (anorexia, bulimia, and compulsive overeating) .

Theory (needs to be verified or not)

Anorexia can be considered to be someone which has a very low lower weight limit.

Bulimia is a strong don't eat message.

Compulsive over-eating is due to an eat message which has high frequency.

It makes sense for an Android to know its

CURRENT WEIGHT

UPPER WEIGHT

LOWER WEIGHT

ENERGY RESERVES (same as FAT)

BACKUP ENERGY RESERVES (Blood sugars/insulin)

Dyslexia and Locale Specific Learning

Dyslexia is a very interesting case because it's to do with audio and visual processing problems related to mapping the HOS visual and audio algebra to the Locale audio and visual symbols e.g. trouble spelling and understanding alphabetic characters for example. What do I mean by the locale? In this specific case, we're dealing with a human in a location and having to learn the teaching methods of that place. In the West we have a phonetic algebra and in other areas it can be symbol based writing with no phonetic basis.

Some people have a HOS Algebra which is not readily compatible/mappable with the Locale; that is to say, they cannot easily map the Locale symbols to their underlying HOS Algebra without, for example, some visual clues. On the surface this appears like a learning difficulty but there is nothing wrong with the IQ of these people. In many cases, they are high achievers and can be more intelligent than mainstream folks who know how to spell their letters for example.

So Dyslexia falls under Locale Mappings to Underlying HOS Audio and Visual algebra.

Therefore we see education in the Classical sense **as mapping Locale information to the person's individual HOS Algebra. Therefore the IQ of an individual in this theory is based on their HOS Algebra abilities and <u>not just their ability to just learn the Locale information</u> such as share language, writing ability, spelling etc; <u>The IQ comes through from the underlying HOS Algebra.</u>**

For an Android therefore, it should have a base Android Algebra and should be taught the Locale specific information which forms a **mapping** to the underlying Algebra. This is pretty important. It is an extension of the language idea where all language is abstracted into a more general form. The intelligence of the Android lies in its underlying algebra.

Spinning, Movement And Balance

The human body appears to have several capabilities around this area. It's key to our mobility.

Inner Ear Balance

Vision based Balance

Predictive movements – imagining

Heel Toe balance

Balance and Co-ordination produces best guesses.

Mainly there are a few competing systems - vision and inner ear balance and mental training vision rooms (it's a good idea not to move eyes too much which can cause disorientation while spinning). Dervishes spin and fix on their arm/hand which is a fixed point in a spinning frame to stop the inner ear messages taking over. People who just spin and their eyes go all over the place fall over as inner ear messages win.

Turning Message (BODY versus HEAD versus GROUND)

Moving Message

Falling Message

Balance Message

Theses are inner ear balance messages and hard to override.

Vertigo/Dizzy is where one has a confused brain. Typically fix on a spot while moving. The virtual self positioning is confused relative to the physical self.

Ice Skater/Ballerina technique is to fix on a spot when they move to keep vision system in charge.

How to spin and keep spinning - brain thinks we're still moving when we are not (are we spinning or is the frame spinning - is the basis of dizziness). Typically, if one spins and keeps no fixed frame, then the inner ear messages kick in and we are forced to the ground for safety reasons.

Heel Toe technique - positioning ones self in the frame, mentally minimizes dizziness with foot messaging for the vision system (Acupuncture toes - eye connection).

Or you can do what a ballerina does by keeping your head in place as much as you can to minimize dizziness.

Toes are related to the eyes/vision in acupuncture so scrunching your toes helps stabilize you.

Ice skaters keep their eyes closed eyes or fix eyes in one position, internally switching into another vision room. This requires mental training where they know their routine in their vision room. Snowboarding has high skill. They go with the routine and try to control their fear by doing the movements and not thinking about mistake. This is similar to a roller coaster, one should switch off ones imagination which is where one's imagination tries to predict all the stuff that could happen.

An Android should have

Inner Ear Balance Messaging

Vision based Balance, use fix point in spinning frame to know it's not ground that is spinning.

Predictive movements – imagining and recorded routines

Heel Toe balance to message virtual self of physical position

Balance Message

Placebo Effect

In this theory, the Placebo effect has a simple enough explanation. We assume that we can learn words and sounds and so on. These are stored as Message Algebra in the brain and the SC does a lookup based on SC Triggers. For example,

NOISE

SHAPE WITH WINDOWS

DOORS

WHEELS

ENGINE

DRIVER

SC LOOKUP

AUTOMOBILE MESSAGING

TYPE

MAKE

HOWSERPOWER

etc;

So certain Visual Triggers produce RESULT MESSAGES. These messages produce a reaction such as

SMILE

EXCITEMENT

DRIVE CAR

These are visual triggers. In the case of medicine, it's no different according to this theory, if one has taken a certain medication or drug or experienced something before the triggers are remembered.

CAFFEINE

result message

SPEED UP REFLEX

MORE ALERT MESSAGING

STORE RESULT MESSAGE PAIR_IN_OUT([CAFFEINE], [RESULT MESSAGES...])

Therefore the SC remembers the reaction of an incoming event and this can include a drink or a type of tablet or a procedure.

Many of the triggers use VISUAL ALGEBRA

BLUE PILL

ASPIRIN

result message

ALTER BLOOD MESSAGES

STORE RESULT MESSAGE

Therefore if we see **what looks like an Aspirin tablet and we believe it to be that our SC does a lookup and produces the result messages.** This is the basis of the Placebo Effect. According to the theory **we must already have the triggers in place and know the resulting messages which are stored in our brains and lookup up by the SC.**

So how can we explain say a doctor curing a Skin disease just by hypnosis. **The patient in question must have triggers and messaging already in place to self-heal.** The suggestion is the triggers. This falls under DNA stored messaging or messaging passes from Parent to Child via Umbilical chord for example.

Visual objects create higher priority messages based on testing

(increasing effect)

TABLET (SUGAR,CHALK)

LARGE TABLET

BLUE TABLET

RED TABLET

CAPSULE

NEEDLE

The degree of messaging is affected by the VISUAL ALGEBRA object. NEEDLE messaging has a higher priority probably because it pierces skin. So we do have the ability to self-heal based on genetic memory but it is SUGGESTION related which just means providing the SC with the correct MESSAGE Triggers. We can also learn how to react to a medicine / substance once we have taken it before because we store the messaging which results from it. So this is just like taking the drink Caffeine for example. Our brain knows how to copy the results of chemical on the brain but we need to send our brain the message.

COFFEE DRUNK

This is only sent to the SC if we think we are drinking Coffee for example.

I am not arguing against taking prescribed drugs or stop drinking coffee! I think the recorded messaging strength decreases over time t so we do need to keep drinking Coffee plus there is a mechanism which enforces the truth of the message so that it must be true that one is drinking coffee! For example, if we feel pain and we produce pain killers, it makes sense that we have really fallen for example. The basic point is that the brain "learns message patterns" and can reproduce them based once it has experienced them a few times. This is the basis of the Placebo effect. The key point is to figure out how for example someone cures themselves of a disease. Their DNA / Body contains the messaging for the cure. They key is to figure out how to reproduce this messaging in others who do not have it.

An Android should also store every interaction result in the same way and figure out the triggers to reproduce the result messaging so it should also support the Placebo effect.

Epilepsy

There are several different kinds of epilepsy. In this theory, we are dealing with problems to do with brain regions which fire the same message over and over again even when they should not. Depending on the Message Type we end up with different symptoms. A MOVEMENT message causes muscle tremors and so on.

There may also be a problem with the SC where it also continuously fires messages and this is more serious as it affects all Message channels. In this case, one may fall over and context switch causing a loss of consciousness and lose total control of ones body. The SC itself may resend messages over and over in the message channel which is not good and cause a general seizure.

Also, it's possible for the SC to stop processing messages altogether and become robot like where one cannot process certain message types and behave as if in a day dream. Here one or more message queues are jammed.

What epilepsy shows us is that different message types end up on different Message Queues so one may have a muscle tremor but still be able to think.

However, if one has a fault in the SC which is roughly located around the Thalamus a person will barely function.

A seizure may also be due to one or more triggers such as lack of sleep or stress but may also be just damaged neurons zones or maybe a biological defect due to a tumor.

So for an Android, we must closely monitor the SC to ensure no Queues are jammed.

If they are blocked we need to keep the messages flowing through.

We must also re-route message centers which are damaged or broken.

It makes sense that the Message generating zones do not malfunction and send the same message over an over again.

High Functioning Autism and Non Verbal Communication

Certain high functioning people with Aspergers' Syndrome can learn and speak technically. They are missing messaging to do with

VOICE_INTONATION_MESSAGING

BODY_LANGUAGE_MESSAGING

FACIAL_EXPRESSION_MESSAGING

These message types are not available to the SC of a High Functioning person so when they do an SC lookup, they are missing the emotion and physical context of human interaction.

This leads to them to speak in a monotone voice and possibly having neutral facial expressions.

However, they can be taught the rules of the missing messaging so they can figure out this messaging by means of a teacher or book because they ability to learn is not affected. However, the missing messaging can lead to an existence where one is literal when it comes to communication.

Get outta here! --- as a joke (missing VOICE, FACIAL and BODY messaging) can make a High Function literally think – I have to leave... which it does not.

For an Android this is REALLY important that it understands and recognizes this type of communication. PRIORITY ONE feature.

VOICE, BODY and FACIAL messaging is required otherwise it will behave like a High Function person which has a very good IQ but is missing this context and emotion. These are the types of Androids one sees on older TV shows.

Multi-tasking and Message Queues

Another important point to make is to do with the ability of a human being to multi-task. High function humans can complain that they cannot multi-task easily. According to this theory, the degree of multi-tasking of a human being is proportional to the number of message queues and the types of messages that they can process simultaneously. A high function according to this theory has less Message Queues. This is analogous to a Computer chip which has multi-core or multi-message processing. So the more message queues the SC has, **the more multi-tasking a being can do. This is a pretty important principle.**

An Android ability rating can therefore be determined by its message queue sizes, types and processing speeds.

Development Coordination Disorder aka Dyspraxia

In the human brain, we have two hemispheres and messages need to flow in and out of the SC Messages queues supporting both sides of the a body at the same rate. One side of the brain supports the opposite side of the body and vice versa. Hand eye co-ordination for writing requires fine motor skills. Tying ones laces requires the use of both hands and eyes in co-ordination with visual memories. Therefore we need Message Queues for both sides of the body to be in synchronization with one another and the major visual and audio centers so that co-ordinated tasks can be done in a timely manner. Therefore, in some cases, the synchronization of the queues is not good and people may be termed clumsy where they might bump into things where one side of the body operates at a slightly different speed than the other.

```
COORDINATION
(
MOVEMENT MESSAGE SIDE A
MOVEMENT MESSAGE SIDE B
}
```

It makes sense according to the design that messaging on both sides of the body operates at the same speed so various tasks can be co-ordinated.

In an Android, therefore one should have millisecond response times which match from both sides so messages are co-ordinated.

Goal Setting, Focus and The Self Activating Reticular System

In the Thalamus where the SC is located according to this theory, there is the Self Activating Reticular System. The idea is that this is a secretary which filters out the messages which are not needed by the SC. So the Subconscious process giant quantities of messaging and the Self Activating Reticular System routes appropriate one to the SC for closer examination.

The idea is that this part of the design is to do with focus and goal setting. For example if one is interested in Topic A and one over-hears this in a conversation one will want to get involved in this conversation and "switch into it". Therefore, the message design pattern is that Message Goals create Message Filters in the Sub-SC which route "messages of interest" to the SC based on the goals of the person for further processing and action. These may be created by the person themselves or produced by the body; for example, a mother with a new born will have goals that listens out for her child crying and ignore a supersonic jet which is louder.

The we enter the realm of motivational speakers who typically try to help people understand their GOAL MESSAGING and set their own goals

For example

GOAL BECOME MORE SUCCESSFUL

Although this seems naive, the idea is that if one sets these goals in the GOAL Center which is the Self Activating Reticular System in the Sub-SC, then our SC will be more focused on it and we will engage more in SUCCESS activities in whatever way we can. The SC can only handle so many tasks at the same time so GOALs are required.

Therefore motivational speakers talk to our goal centers, help us understand them and set new ones.

So

Sub-SC **MILLIONS OF SUBSCONSCIOUS MESSAGES SOUND, AUDIO, VISUAL etc;**

SUBCONCONSIOUS MESSAGES, FILTER BY GOALS

produces (Send to SC for Further Processing)

SOUND CHILD CRYING

NEW BUSINESS OPPOTUNITY

NEW CAR TYPE ONE IS INTERESED IN

So the GOALS produce FILTERS in the Sub-SC which are routed to the SC.

An Android should therefore also have dynamic and system goals and the equivalent of a Sub-SC which handles millions of message automatically, for example with core System Functions. Using this approach we can allow an Android to “specialize” in a particular activity such as being a Rock Specialist on a Foreign World so that it spends its time doing this type of analysis and enjoy it. Goals that are accomplished produce reward/ feel good messages.

Chapter 7 - The Olfactory Senses And The Brain

We'll delve into the brain structure here and how we smell and taste.

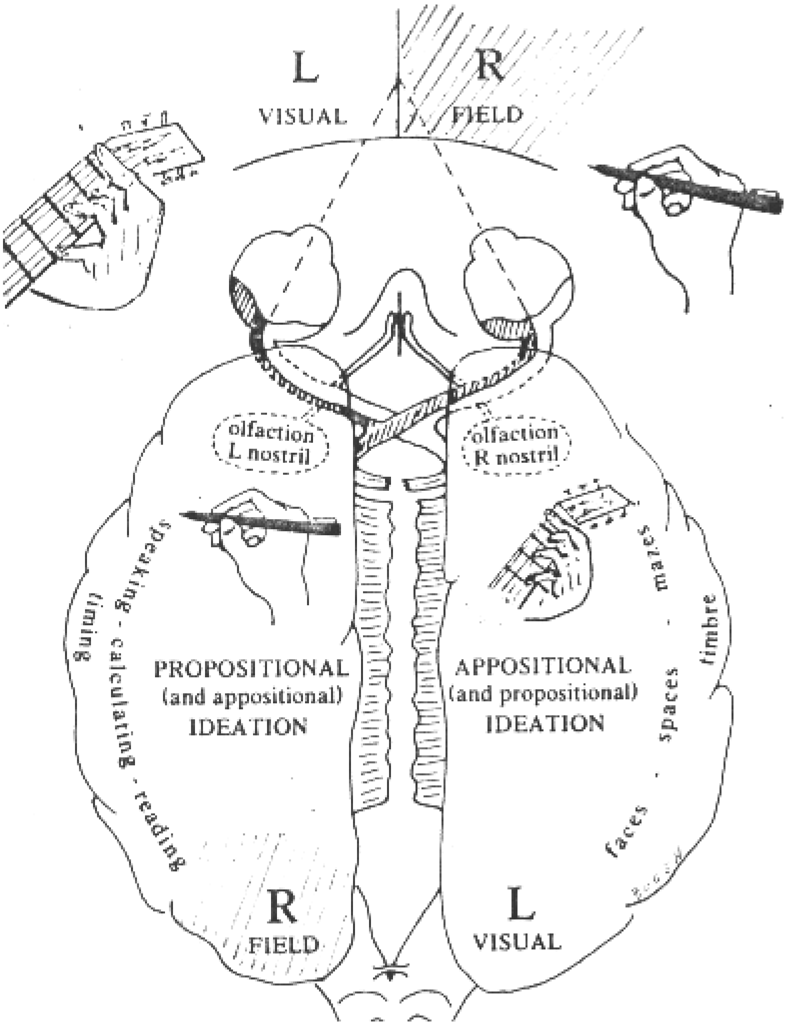
Two Brains Are Better Than One

Human physiology shows us that we have two brain hemispheres joined by the Corpus Callosum. We also have two eyes and ears and so on. Therefore it's been a design decision by our DNA to provide backup of our various essential senses.

An Android should copy this design pattern. In theory one can have many ears and eyes in an Android device but the design pattern should be to have at least two so there is a secondary system in place.

We also have the idea of cloud computing so messages can be stored remotely as well.

Split Brain And Dependent Neural Areas of Specialism


L
R
VISUAL
FIELD
olfaction L nostril
olfaction R nostril
speaking - calculating - reading
timing
PROPOSITIONAL (and appositional) IDEATION
APPOSITIONAL (and propositional) IDEATION
mazes
timbre
spaces
faces
R
FIELD
L
VISUAL

In some medical procedures the Corpus Callosum is severed and a person ends up with two independent brains. Also someone may be born without the Corpus Callosum and there is independent thinking on both sides.

Examples are

Child grows into adult and demonstrates a lack of understanding of emotional context of language or facial recognition. We talked about this before and it emerged that the person does not get Emotional Messaging. Therefore the related messaging cannot be computed by the different regions, I'll explain more shortly.

Second example is a person who at a certain point in their adult life has a procedure to cut the Corpus Callosum in half which stops Message Algebra from traveling from one side of the brain to the other.

Example is (severed Corpus Callosum)

Word “CAT” appears on left side

Patient looks and says “CAT”

Word “CAT” appears on the right-side

Patient looks and says that they cannot say it

However they can draw a cat

First point to make according to this theory is that **specialized area of the brain have related neural pathways.**

For example

CEREBELLUM_NEURAL_CONNECTION_SHARE_MESSAGES(VOCAL_PROCESSING,IMAGE_DRAWING_PROCESSING)

This is a two way connection which can share either way.

In the human brain typically these are on different hemispheres.

So let's work through the example according to the theory.

Patient sees “CAT” on left side

Core (Legacy “old” Brain) does Vision Pattern Recognition which ends up in Sub-SC Queues

Sub-SC has a FOCUS GOAL “Cat” which sends messages to SC

SC which has left side and right side (aka Thalamus) passes MESSAGE Algebra “CAT” to Cerebellum on the left side.

Left side Voice Processing Center gets “CAT” which ends up in the left hemisphere

Person says “Cat” no problem

In a healthy person with an inter-connected brain, the Message Algebra is sent to Image processing center in the other brain hemisphere as they are “paired”. DNA determines the pairing associations, so one can draw it and say it.

In the other case where the brain is split, the message goes to the other side and one can draw a Cat but cannot say it as there is not interconnection.

Therefore we can see

SC works for each Cerebellum side as they are paired.

The specialized areas themselves “share” related messages.

Cutting the brain in two prevents message sharing between related Message Specialist Areas.

An Android should therefore have a list of Specialized Message Zones (Sentence Construction, Vocal Abilities) and their Related Messaging Pairings. The results of their work end up on the Message Bus and go back into the SC. Placing areas of specialism on other hemispheres is probably a good idea.

Olfactory Senses And Taste

Ones sense of smell is closely related to ones ability to taste. It's been shown that if one loses ones' sense of smell that the emotional context of food is lost and that the taste is lost.

The tongue handles taste messaging for

SWEET

SOUR

BITTER

SALTY

which are sent to the Brain Messaging center for this and then go to the SC.

The olfactory centers in the nose which handle what are called volatiles are sent to the Hypothalamus and pairs with emotional messaging.

So if we eat something, we produce

Taste Messaging

Emotional Messaging

Olfactory Messaging

which in turn ends up in the SC

Emotional messages are looked up and paired with

Memory messages

So the act of eating a favorite food produces a feel good factor and possibly memories of earlier times in ones' life or not as the case may be.

The Olfactory messaging itself is the first sense which we developed before our DNA had vision and hearing so eating is a form of looking up the relevance of the food, where and when to find it, and is key to survival. Therefore we are richly rewarded for eating good food by our bodies.

Up to two percent of human DNA is dedicated to the olfactory senses to figure out what food is good and what is not. Sweet and Fatty foods are favored by the brain because they have high calorific value. However, the volatiles in tomatoes for example may make a tomato seem sweeter to us even if it is not sweet and may smell completely different.

Therefore our concept of a sweet food is really more a brain messaging calculation for nutritional goodness.

So

SWEET_MESSAGE is an amalgam of taste factors. It is not necessarily sugar per se.

We also have strong messaging for preferred foods and learned foods based on our family foods.

SOUR_MESSAGES are really just a body's way of rejecting certain types of foods based on the brain calculations.

The fact that these messages end up in the hypothalamus and are paired with our emotions and memories indicate according to this theory that this sense developed before sound and vision which are paired in the SC. Our body tends to remind us where we found the food and makes us feel emotionally good if the food is good for us, making eating a very enjoyable experience. This makes sense.

According to this theory, the nose is a part of our senses which we underestimate in that it handles not just food but subliminal inter-group communication between other people.

An Android should have the ability to handle scents like this also. It makes sense that it can also taste like a human in order to understand the value of food and taste and recall where it is sourced and how it is cooked.

CHOCOLATE

VANILLA

etc;

Pheromones And Non-Conscious Inter-human Communication

The most obvious way we communicate is visually with words and sounds and the way we dress etc; However a very powerful old method of inter-communication between humans is via our sense of smell. It is such a basic sense and an old sense in that it drives us to make choices without realizing it. Clearly if one smells bad, people will literally evade that person without much thought. The feeling is over powering and one just does it. This is because it's an old sense and the newer ones are tied to logic and reason. It's a natural thing to prefer to know why we make decisions and more reassuring.

BAD SMELL MESSAGE

EVADE

In the human olfactory center there are two centers. According to this theory, one center handles our sense of smell related to food and the second one handles inter-group communication.

When a mother holds a baby she will smell the baby and the baby will relax in the presence of the mother's smell.

Women who live in close proximity find their reproductive cycles synchronize due to scent messaging between them.

People who wear aftershaves with pheromones find they are more accepted and liked in social situations.

The reason for these examples working is the power of certain olfactory scents which are very old. According to this theory, there is a vast range of scents which affect human beings in all kinds of ways we do not even realize. This is very similar to Acupuncture except that these messages are airborne. I do not have a full list of them but the theory indicates that they exist. This of course relates back to the perfume industry and how the scent of a good perfume makes someone smell "good". The idea here is that the brain turns the perfume smell into a positive message in the brain.

On an interesting side note, if one imagines traders on a stock market floor and it's true that human beings emit a “fear scent” which for example dogs can smell as they have a more developed sense of smell, then it's theoretically possible that other traders smell this scent subconsciously and will sell stock possibly causing an accelerated sell off of a stock. An experiment would need to be created to prove this idea one way or another.

An Android should also support airborne scent messaging. It should categorize the scents into different types.

TOXIC

PLEASANT

GOOD

etc;

Brain Architecture And The Limbic System

The core architecture of the brain messaging according to this theory is located in the Limbic System.

The SC is the Thalamus which processes goals and has the parallel message queues where we have “Consciousness”

Note: Thalamus = Descartes = I think therefore I am. Also the virtual self is located in the visual centers whose location I will cover later.

The Sub-SC is beneath the Thalamus on the brain-stem where one has “focus” and filters out messages for the SC (audio, video, etc;)

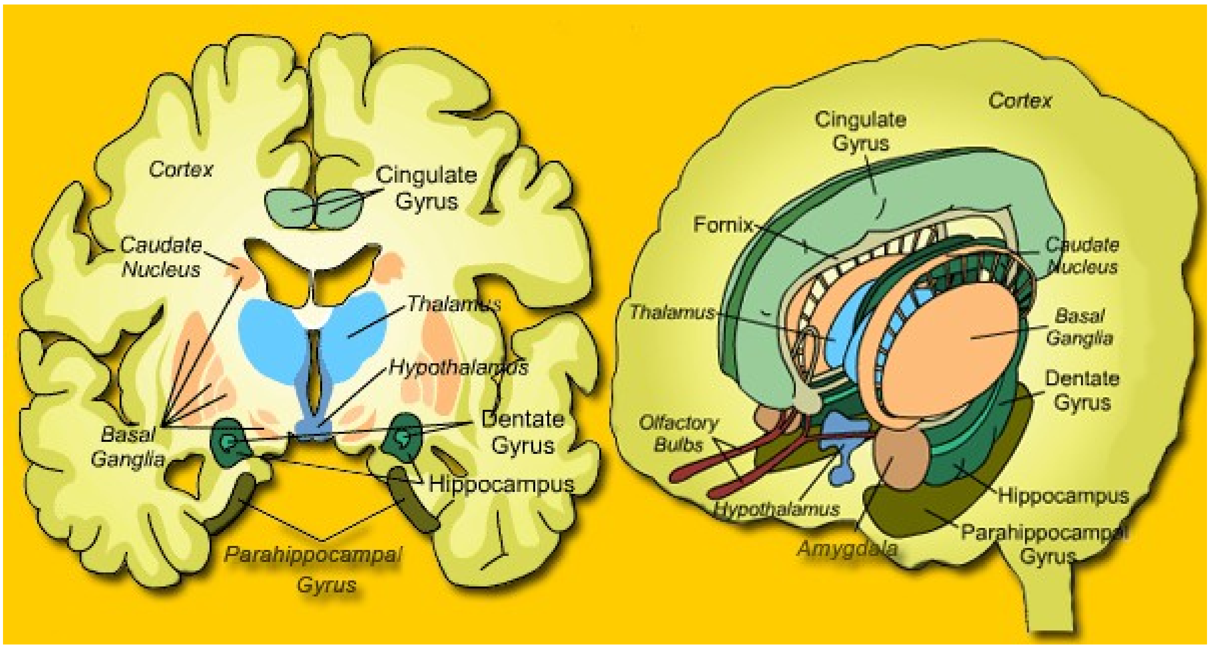
Memory lookup via messages is in the Hippocampus

Thalamus lookup of memory is covered by the Fornix which is a message bus also supporting the Hypothalamus

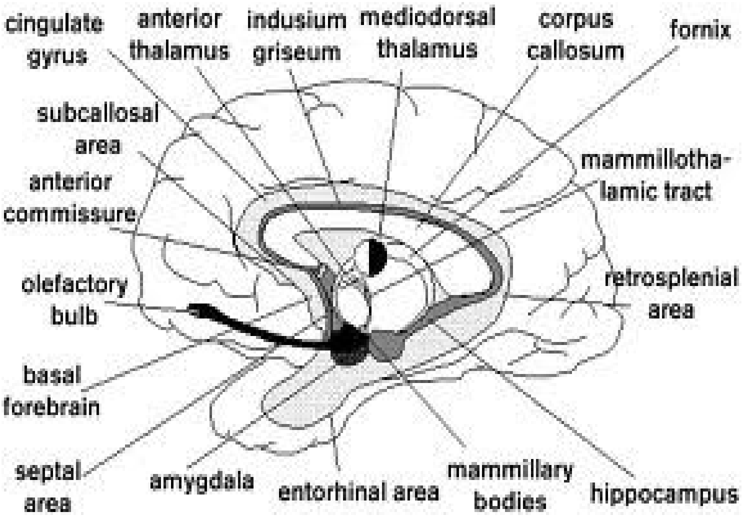
Importantly, the second olfactory sense (e.g. pheromones) operate outside of what we think of as Consciousness which is in the Thalamus, so we can be influenced by certain smells non-consciously via the Hypothalamus.

Also the Thalamus covers what we think of as “our humanity” with logic and reason where we receive messages from the Pre-Frontal Cortex and the main brain message channels.

And there are two identical systems which inter-communicate at the Cerebral Cortex level via the Corpus Callosum



Breaking down the limbic system



An Android should also implement an Architecture like this. It's preferable that the Olfactory sensations are detected by the Android unlike in a human and will require some kind of dedicated message queue.

Aphasia

People who suffer from aphasia can for example try to speak but it comes out making little or no sense. Also, they can suffer from sentence aphasia where the sentences make no sense. The way this works according to this theory is that the Message Algebra is transmitted to the speech or sentence centers. The learned mappings since childhood are damaged and therefore the sentences or sounds makes no sense. However, the person knows what they are trying to say. Therefore, this shows that we have an underlying Algebra for communication which is independent of the centers that do real world mappings.

An Android should have an underlying Algebra before a locale is mapped to speech/language etc;

Amygdala

This is the old emotional system which is hard coded in the older part of our brain. It is responsible for

FREEZE

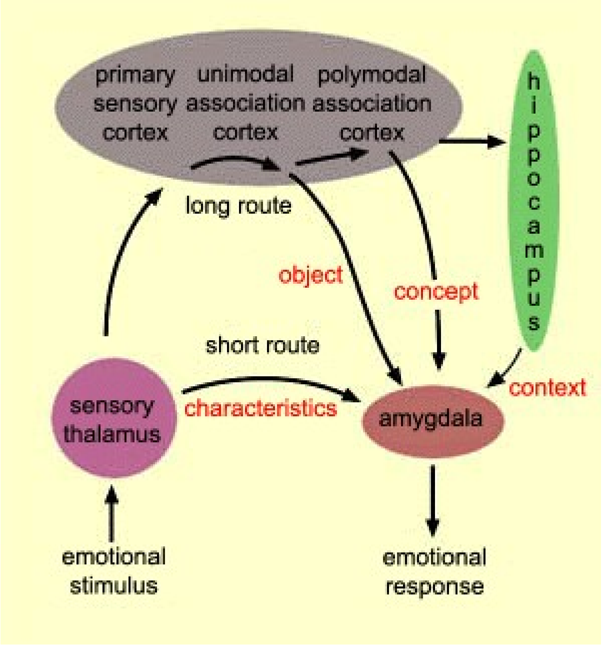
FIGHT

FLIGHT

ANGER

etc;

It ignores the Cortex for high level threat messages from the SC.



The short route would be a high level threat such as message

e.g.

CHASED BY TIGER

We can also become emotionally hijacked by this part of our brain which is the “fast action” center. High priority danger messages are routed to this part of the brain by the Thalamus for example if one is being chased by a Tiger. There is really not much to think about other than move somewhere safe.

This part of the brain makes us feel very alive in that it increases our metabolism and everything seems very vivid but the memories may not in fact be very accurate. People engaged in conflict may use this part of their brain a lot.

However one can suffer from Road Rage and in this case, we need to exercise Emotional Intelligence. In this case, we need to gauge the level of real threat to us – apart from a hearing a honking horn of another car (I know this feeling)! Therefore, the PreFrontal Cortex sends up Logic and Reason messages which can override the high level messages but it's not easy the body can become flooded with Adrenaline so one needs to control breathing and so on in Road Rage cases.

e.g.

CALM MESSAGE

LOWER THREAT

Also, if one's Thalamus / SC personal goals are not reached due to a situation, emotional Messages can be generated such as

ANGER message

FRUSTRATION message

where one feels “thwarted”. This end up in the Amygdala and we enter the world of Anger Management where the person needs to engage their PreFrontal Cortex more to moderate and neutralize these ANGER messages. An interesting case was where a worker damaged the part of their PreFrontal brain where Anger control messages were generated and their personality changed and they became angry a lot.

An Android should also have high level threat messaging and response protocols also and these need to be carefully implemented. The threat response logic must be carefully implemented in the Android and should probably remain calm most of the time compared to a human.

Chapter 8 Vision

Here we will cover how we begin to see the world and define the requirements.

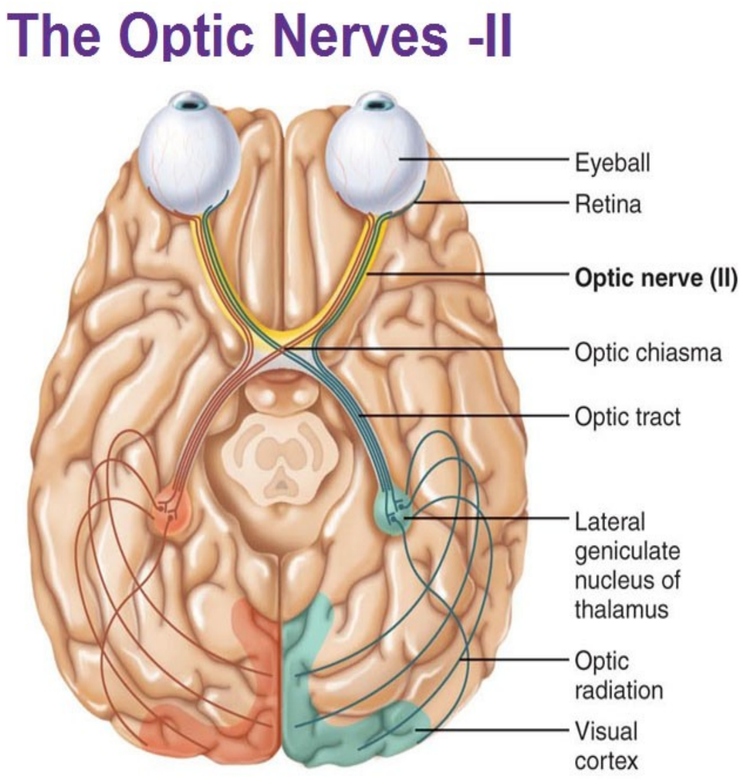
The Graphics Card In The Brain

According to this theory, what we see from one second to the next with our eyes is really a brain generated image based on VISUAL ALGEBRA messaging from various parts of our brain. The first and most immediate question to ask is where is this virtual view of the world generated in the brain? According to this theory, the **Thalamus has the equivalent of a Graphics card built into it and it processes Visual Algebra messages to build our world view.**

The diagram points to the optic nerve Lateral geniculate nucleus of thalamus.

In this region Visual Algebra message are sent and the “Window in the Mind” is created in real time.

Both Thalamus' vision systems inter-communicate to build the joint view with depth and so on. More on this later.



In an Android, the “graphics card” is replaced by a physics engine of some type. Theoretically the Pi-Space physics theory could provide the physics components for it but does not have to be. The engine should be message based where it builds a world based on its “best guess”. Therefore the scene is built with the available information supplied.

VISUAL MESSAGE OBJECT MOVEMENT X

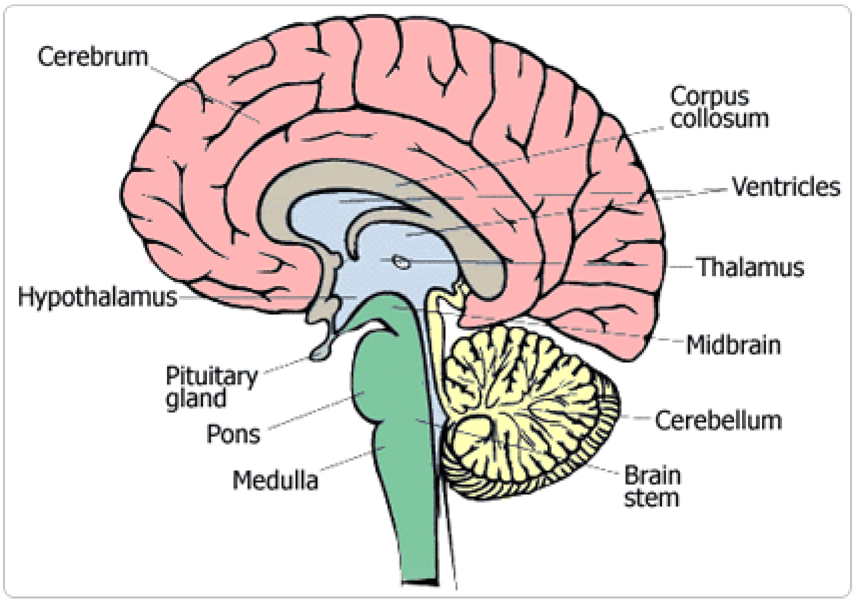
VISUAL MESSAGE OBJECT MOVING CLOSER Y

The engine then sends out collision detection messages like

VISUAL MESSAGE AVOID ONCOMING OBEJCT K

The Virtual And the Physical and the Cerebellum

It's all good and well to have a nice real world view in ones mind where one is placed so that one can determine the next steps but this needs to be mapped to the real world and these two mechanisms need to be closely coupled so that the virtual view is an accurate copy of the real world. In the human body according to this theory, the part of the body which handles reading from and writing to the physical form is handled by the Cerebellum. This is then synced with the virtual view in the Thalamus via messaging. The Cerebellum is called the "small brain" and is the Physical world synchronization mechanism.



The two systems need to be closely coupled and are. Both systems are pretty core to elementary function and the tree like structures feed nutrients to the cells in both the Thalamus and Cerebellum which is a bit like a computer system with large power requirements. The Cerebellum figures out how accurate the vision system is and if hand-eye co-ordination is off due to poor vision it applies offsets and the remembers them. This way motor function is remembered automatically via lookup mechanisms

MOTOR FUNCTION VISION OFFSET X,Y

LEARN MOTOR FUNCTION SEQUENCE X[

MOVE HAND

MOVE ARM

etc;

]

TRIGGER OPERATION A, IMPLEMENT SEQUENCE X

An Android should also have the equivalent of a Cerebellum for co-ordination with the Virtual View and the Physical world and have the ability to Learn Motor function steps.

Very High Level Vision Design

The high level design is pretty simple. Implementing the fine detail is very difficult and for this reason about one third of the newer part of the brain cortex is taken up with various visual processing tasks.

So the high level pieces are as follows

Sensory equipment to view the world. In the human case we have two eyes, in a spider many.

A VISUAL ALGEBRA capable of supporting all visual requirements. This is pretty large set of messages.

A piece of Bio or other Hardware capable of turning all this Visual Algebra into a Window In the Mind which appears “Real”. This is pretty large.

A SELF LOCATION ALGEBRA which positions the Virtual Self in the Window In The Mind or other Window. This is mid sized.

e.g.

LOCATE SELF AT X,Y,Z

SWITCH ROOM TO DREAM ROOM

SWITCH ROOM TO REALITY ROOM

A CONSCIOUSNESS ALGEBRA which allow the Virtual Self to Think. This is related to the SC. This is already covered in some detail so far based on ones' goals in the SC.

We also need (sub-specialized) vision brain regions for

Sub-Specialization of VISUAL ALGEBRA

COLOR
MOTION DETECTION
OBJECT / FACE RECOGNITION
STEREOSCOPIC 3D
OBJECT SIZE DETECTION
etc;

We also have the requirement that this happens in real time, typically in a couple of milliseconds of less.

Therefore each brain region communicates via agreed upon Messaging protocols which are defined in the Messaging Protocols within the DNA.

We also need to pair the VISUAL ALGEBRA with AUDIO ALGEBRA as discussed earlier.

The Graphics Card in the SC therefore will build a world view with whatever it gets in real time, time slicing message every 'n' milliseconds. For example, if a person has suffered a stroke in the region of the brain for Color Processing, the SC-GC (System Consciousness Graphics Card) will build a world view without color and keep working. Some stroke patients experience this but can see and operate reasonably normally.

Let's introduce some terms

SC-GC = System Consciousness Graphics Card (it builds a 3D world view based on Visual Algebra)

SC-RR = System Conscious Reality Room (The room we live in when we are awake which is built by by the SC-GC)

Note: There are many “rooms” that that SC can live in. For example we might day dream in another room in our mind.

An Android should implement a design like this.

Taking Visual Snapshots And Assembling Them Into an Overall View using Saccades

The way that the human eye works is that it is constantly working jumping from one place to another in a fraction of a second and focusing on it. If we move a camera backwards and forwards like this, then we get blurring.

The way the human eye works is that the image processing is switched off when the eye jumps from one location to another. The brain then reassembles the images to form the SC-RR based on the visual algebra once the eye is stationary on a point.

e.g.

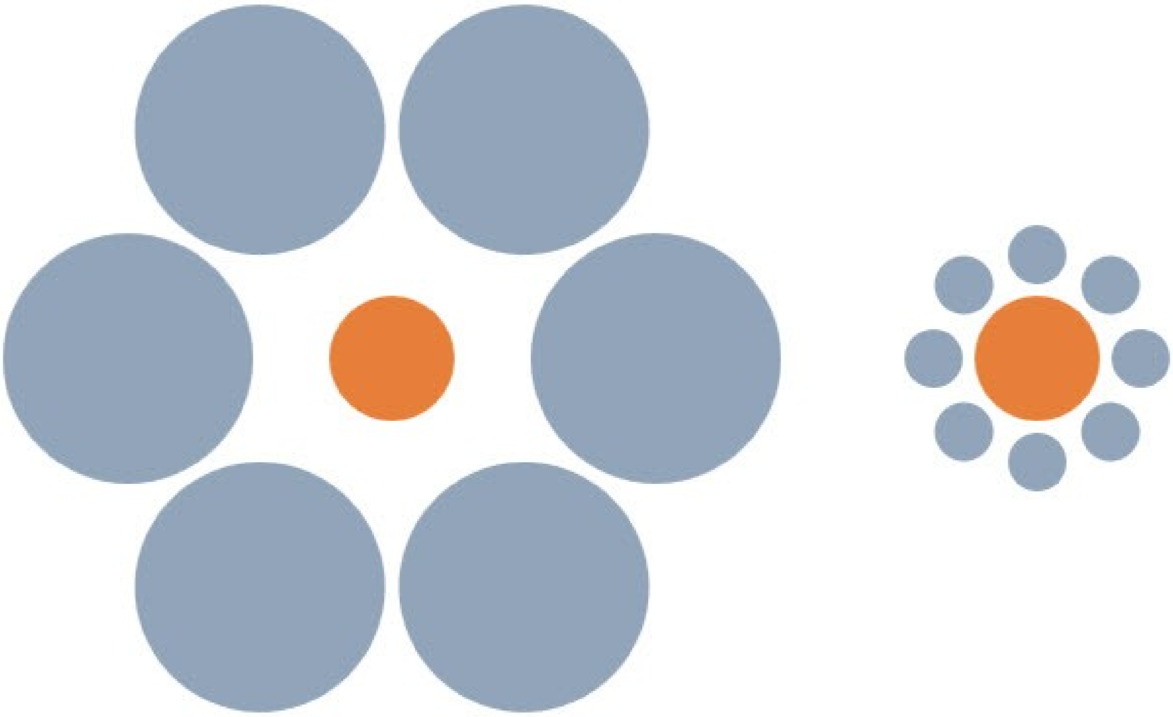
VISUAL ALGEBRA MOVE EYE TO SCAN LOCATION X,Y DISABLE PROCESSING UNTIL DONE

An Android should work this way as well.

Depth Perception and Size of Objects

According to this theory, the brain calculates the sizes of objects based on visual cues. We all know from logic that closer things are larger and further away objects are smaller. However humans are not equipped with distance sensing radar so the brain applies an algorithm which figures out distance and size the best it can. Therefore if lines converge we know things are moving away and smaller. Also, the relative sizes of other objects help us figure out what is closer and what is further away. **Therefore the VISUAL ALGEBRA adjusts the sizes of the objects to make them relatively larger or smaller depending on the visual context in the SC-RR. All SC-RR rooms are three-dimensional so object position is important.** The object will appear larger even if it is not physically on the flat screen when we measure it. According to this theory, the **object <u>is larger</u> in the SC-RR where we visualize reality in a 3D context.**

Ebbinghaus demo



The inner circle on the left is the smaller one in the SC-RR. However if you measure them they are the same. We get the impression that the one on the left is further away. The depth perception in the brain work this out and sends VISUAL ALGEBRA to the SC-RR

e.g.

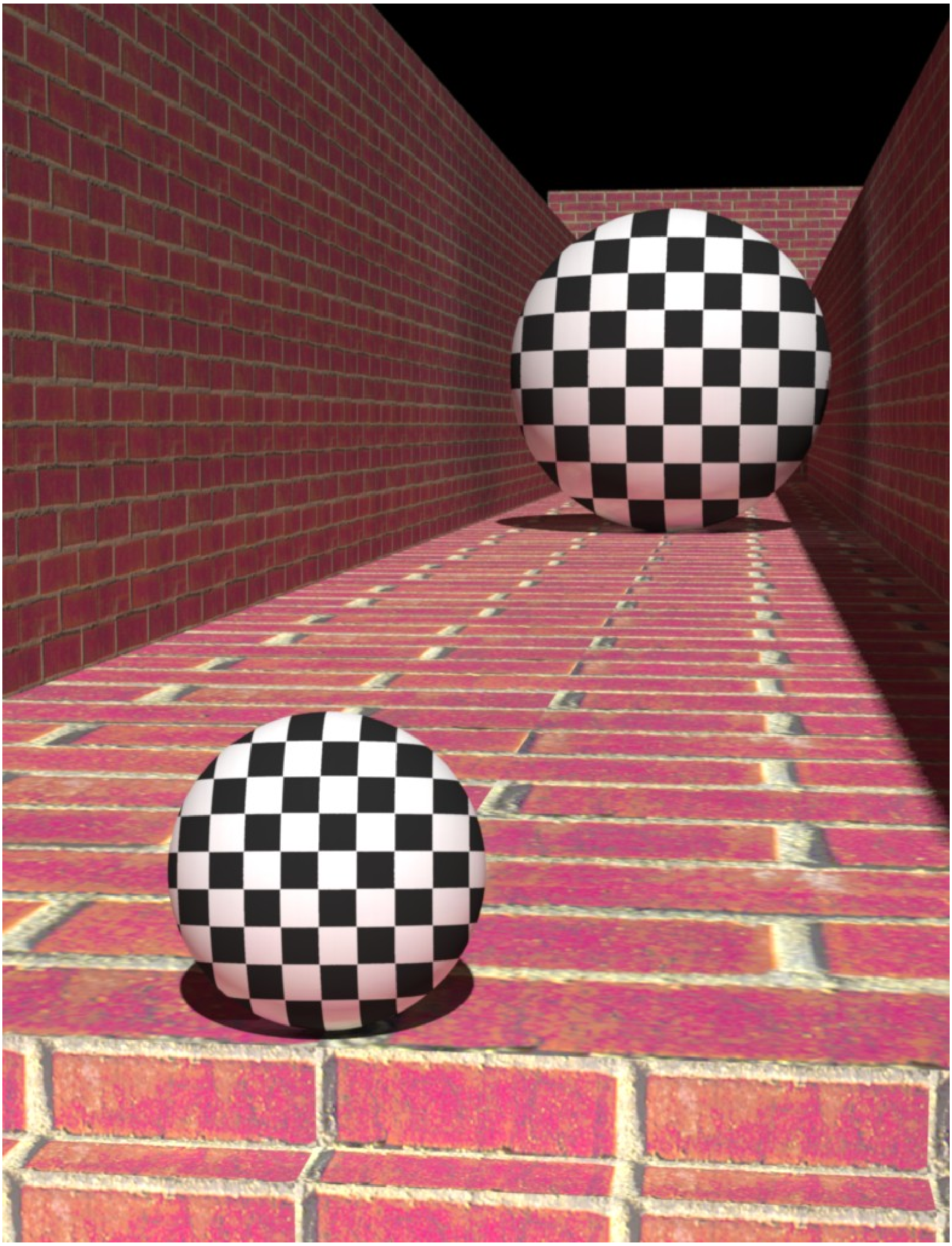
VISUAL ALBEBRA INNER CIRCLE LEFT-SIDE RELATIVE DISTANCE 2 SIZE 1

and

VISUAL ALBEBRA INNER CIRCLE RIGHT-SIDE RELATIVE DISTANCE 1 SIZE 1

Another nice example are the two spheres example. They are actually the same size on the screen but the brain makes the further away one larger relative to the walls.

It is larger in the SC-RR but when we measure it on screen/paper, is is not.



VISUAL ALGEBRA CLOSE SPHERE RELATIVE DISTANCE 1 SIZE 1 and

VISUAL ALGEBRA FURTHER AWAY SPHERE RELATIVE DISTANCE 10 SIZE 3

A low moon on the horizon also appears larger even though it is not because the brain sees buildings and sends VISUAL ALGEBRA OBJECT A DISTANCE messages making it seem further away and therefore it must be larger relative to the buildings.

Therefore the depth perception algorithm identifies all the relative objects in a scene which is by default 3D and makes best guess about relative distance and size. The SC-RR image is then created.

An Android should also do this. It requires that Object detection and image analysis is done first.

Macular Degeneration and The Mind's Eye

People who suffer from Macular degeneration or loss of vision often report that they can see things in their Mind's eye which are not there in reality. Alternatively, their failing vision adds things like colored squared fur, objects etc; Many patients fear that they are losing their minds. This was discovered in 1760 by a doctor whose father suffered from this.

This is actually reasonably clear to explain if one assumes we have a SC-RR or another room which accepts VISUAL ALGEBRA messaging. In the case of a mind with failing vision, the external world messages fall off and are replaced in about ten percent of people by brain produced VISUAL ALGEBRA messages. This is analogous to a system test mode for a part of the body not receiving expected stimulus.

Some people finds these VISUAL ALBEGRA messages enjoyable but others do not. Best guess here would be to try to get them to do mental exercises where they can try to control the visual messaging.

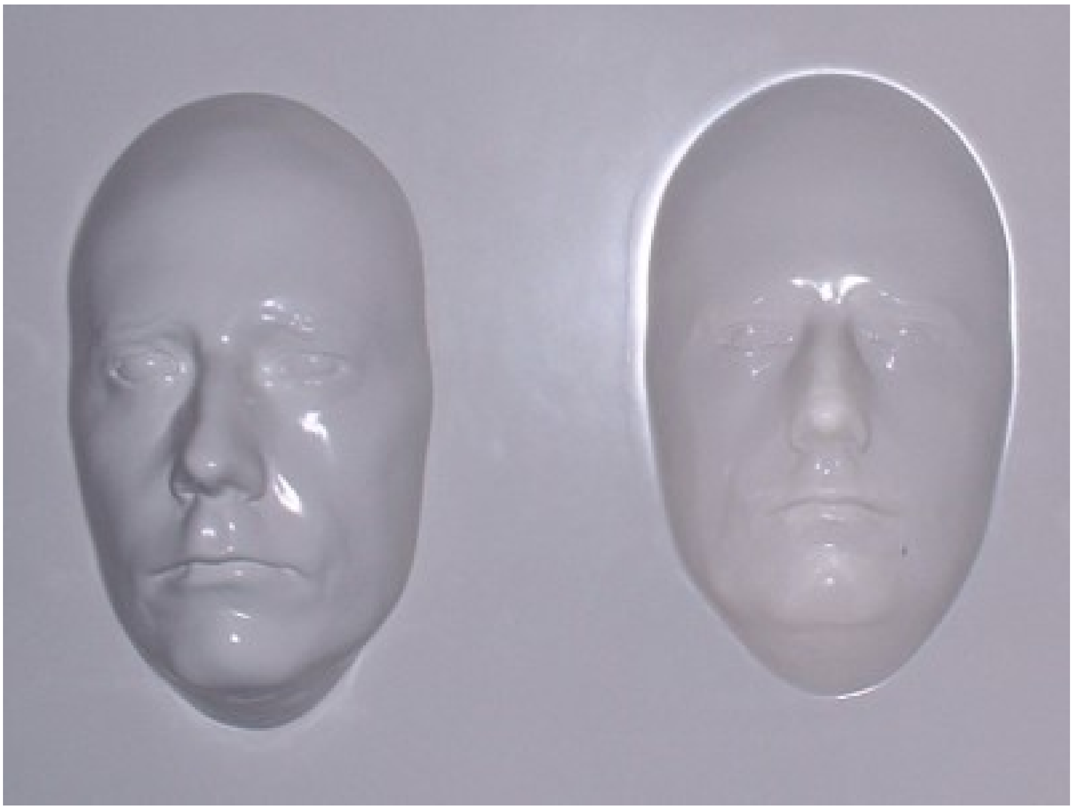
An Android probably does not need this but the idea of placing VISUAL ALGEBRA messages in any arbitrary room or on top of a SC-RR is definitely useful.

Facial Recognition

A human face requires a specialized region to process it. It analyses the various parts of the face and has a "face database". It measures each of the properties of a face. When this information is passed to the SC-RR, the face extrudes.

An interesting experiment shows how an inner shadow face points out to the mind because this is the way that the SC-RR builds it.

The hollow face on the right points in but the brain changes it.



An Android should also use shadows and light to figure out its own "face database" so that it can recognize one person from another.

e.g.

VISUAL ALGEBRA

FACE NOSE

FACE EARS

FACE EYES

FACE CHEEK

FACE CHIN

FACE LIPS

FACE BROW

FACE HAIR

The Optic Nerve

The optic nerve demonstrates how the brain scales but giving each nerve its own fiber which runs to the Vision processing center at the back of the brain. There are between 770,000 and 1.7 million nerve fibers.

An Android probably cannot have a dedicated wire but can have a dedicated channel for each vision signal. To scale this to several million pixels will be tricky but an Actor model is required.

Back lighting a Visual Scene

A nice technique for a visual illusion is to look at two different blocks of color for thirty seconds or more. Then look at say a desert scene and it will be colored with a brain color filter, different on either side of the brain vision system. Therefore the brain is always trying to figure out the back-lighting and may alter a foreground color such as a circle of color.

Therefore we have

VISUAL ALGEBRA BACK-LIGHTING COLOR

VISUAL ALGEBRA FORWARD COLOR ADJUST USING BACK-LIGHTING

This is important because the brain needs to add these two effects together to get the best “survival picture”.

This can also create a Color which looks like another one until it is taken off the back-lit color and then it will appear to change color!

2D Messaging versus 3D Messaging

According to this theory, the brain supports two types of VISUAL ALGEBRA for vision.

2D Visual Messaging

3D Visual Messaging

e.g.

VISUAL ALGEBRA

2D SQUARE SIZE 16

3D CUBE SIZE 16 DEPTH 20

3D SQUARE SIZE 11 DEPTH 18

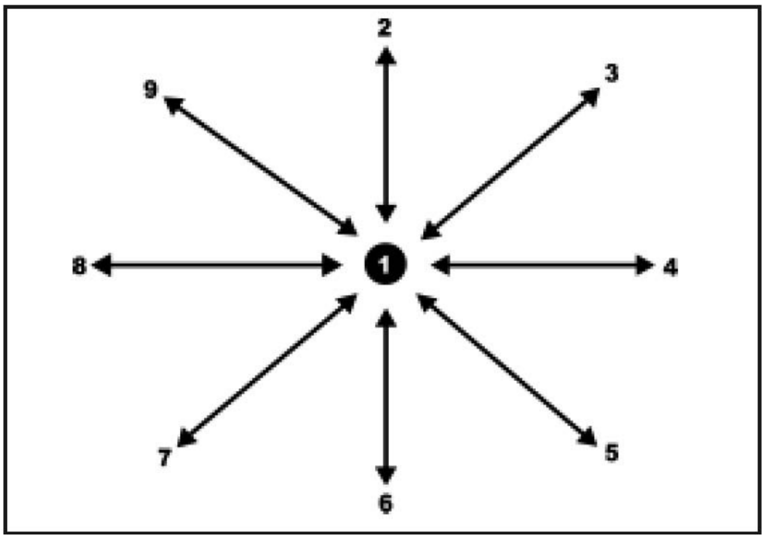
The 2D messaging produces “cartoon like” imaging and the 3D produces the Stereoscopic images we call “normal vision”.

The argument in favor of this viewpoint is that some people who lose their vision see “cartoons” which are 2D messages.

For now, I will not go into how the brain produces 3D messages, suffice to say the brain supports it.

The Eye

We need messaging to handle the different functions of the eye and its positions.



Plus there are different behaviors of the eye such as tracking and blinking and so on.

Messaging examples.

Eye Move x (cardinal directions)

Eye Move y

Eye Focus

Eye Retina

Eye Peripheral

Eye Light Adjust

Eye Blink

Eye Smooth Track

Eye Partial Track

The eye has rods and cones. Two images overlap and are merged into one. Each image is sent down the eye on individual optic channels. The first stop is at the Thalamus LGN and then it goes into the Visual cortex.

An Android should have this as well. Two eyes which synchronize.

Specialized Neuron Groupings

The brain contains specialized Neuron groupings. Many have the job of pattern recognition in the Visual Cortex. These are built into our brains by default and is part of our Natural Visual Algebra.

The Neuron groupings look for specific patterns in particular orientations its been shown by experimentation.

In the visual cortex of the brain, there is an area called V1 which receives the Visual information. This area is capable of finding lines and curves within objects for example and it sends information like this to the next part of the brain for visual processing and it does this for tiny regions of the brain.

e.g.

CURVE TYPE A SIZE 2 FOUND AT X,Y

LINE SEGMENT TYPE B SIZE 3 FOUND AT X,Y

There are millions of these basic Neuron shape detectors firing if patterns are matched and they are added up in the next specialized region.

The brain has natural algebra for SQUARES, CIRCLES and so on in its Algebra. An area like V2 will then detect higher order shapes in a 2D Algebra.

SQUARE SIZE 2 AT X,Y

CIRCLE SIZE 3 AT X,Y

and so on.

An Android should do this. To do this requires lots of parallel processing which uses the Actor model.

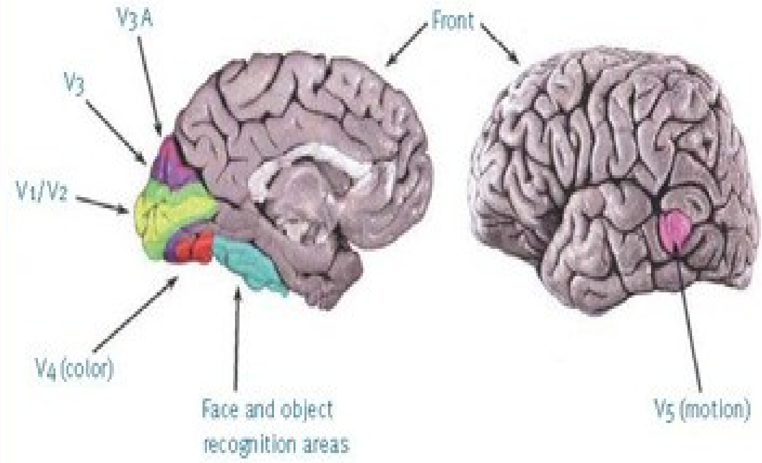
Aggregating the Visual Algebra to recognize patterns in V1

The LGN routes the trans-ducted visual information to a region called V1. This does base pattern matching for simple patterns. These are then aggregated to V2 and so on where more information is determined about the image. This happens in a fraction of a second.

Each shape has a range of values.

According to this theory, these map to VISUAL ALGEBRA 2D MESSAGE TYPES.

Letters are “learned” shapes whereas circles, squares and so on are part of the default algebra and do not have to be learned.


V3 A
Front
V3
V1/V2
V4 (color)
Face and object recognition areas
V5 (motion)

Base Neuron Pattern Recognition e.g. V1
shapes identified

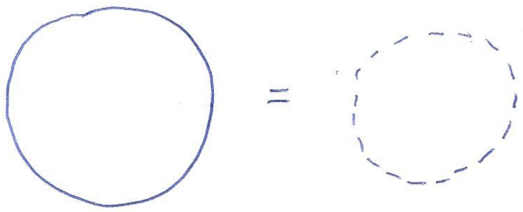
— \ | / — \ | / / —

Each pattern has a range of angles
e.g. — to \ | >

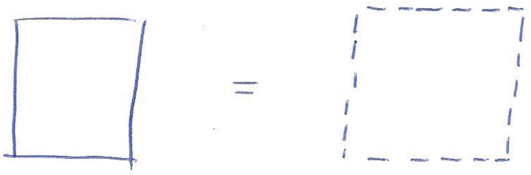
If a match is found it fires Visual Algebra

These are aggregrated and sent to V2 for higher an alysis

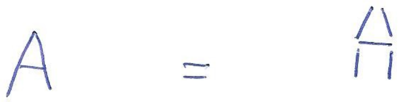
e.g. circle



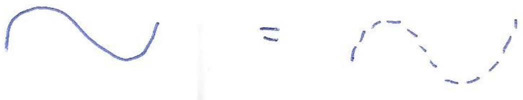
square



Letter



Curve



An Android should do it this way as it makes a lot of sense.

What Is It? Where Is It? Dorsal And Ventral

The visual cortex has two main circuits. On one circuit, the task is to figure out what one is looking at. I've already alluded to that with the pattern recognition. The other aspect of vision is the ability to have “hand eye co-ordination” or “where is it?” where one reaches to something and picks it up for example.

The Ventral System is “what is it?”

The Dorsal System is “where is it?”

It's commonly called two circuits related to vision.

The Ventral system produces the 2D and the 3D Visual Algebra for the SC-RR but it's mostly a static analyzed view of the image.

The Dorsal system syncs the Virtual SC-RR Image with the Physical body based on real world goals so this requires message interaction with the body proper.

The example case is where one has a damaged Ventral System, one can reach for an object and fetch it but one cannot describe the size of it for example. Therefore, some of the Visual Algebra is missing for the SC-RR objects.

The other case is where one has a damaged Dorsal system, one can describe the object to be picked up in terms of its size for example but one cannot synchronize the hand eye movements for the to grasp the object.

For this reason it's claimed that there is no window in the mind just specialized area which makes sense. However, this theory adds to these ideas that these messages are recombined.

The way this is explained in this theory is that a fully functioning Dorsal and Ventral system feed Visual Algebra back into the Thalamic Nuclei producing a fully functioning SC-RR. I'll show this shortly how I work this out.

Let's take the example goal in the SC of

FETCH CUP

The Dorsal system produces MOTOR, EYE messages like

ARM MOVE RIGHT

REACH HAND

MOVE EYE

ARM MOVE DOWN

OPEN THUMB

GRAP HANDLE

and so on

The Ventral identifies Objects with Visual information

VISUAL ALBEBRA

OBJECT IDENTIFIED CUP SIZE 2 DEPTH 4

However, the SC-RR knows it is a Cup for a damaged Ventral system example but not much else about it e.g.

VISUAL ALBEBRA

OBJECT IDENTIFIED CUP

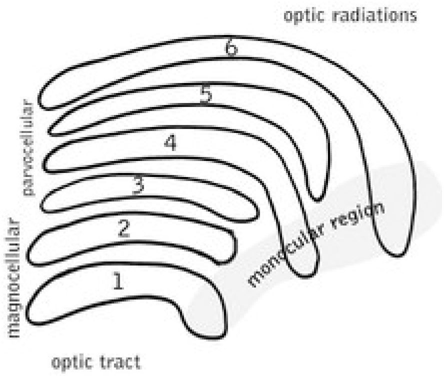
Also, the Ventral and Dorsal messaging information is broken up in the LGN as I'll show shortly.

The reason for the two circuits is that it requires different specialized neurons. An Android should use this idea also. Co-ordinating hand-eye in the SC-RR requires a specialized set of messaging (Dorsal).

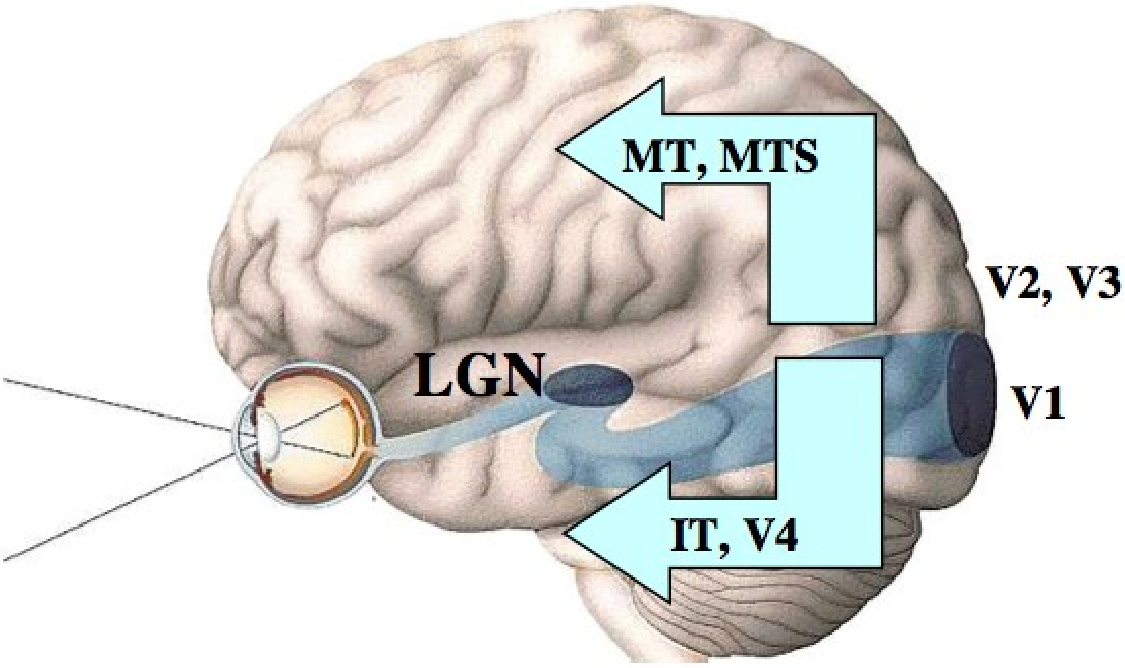
Routing of Where, What And the LGN

The optic nerve routes the visual information to LGN which has several layers and these determine the where,what routing. Mostly, this is a human, primate layout view of the world in the LGN. Research shows these mapping are different for other species so this is our specific architecture.

LGN



Overall the wiring is as follows.

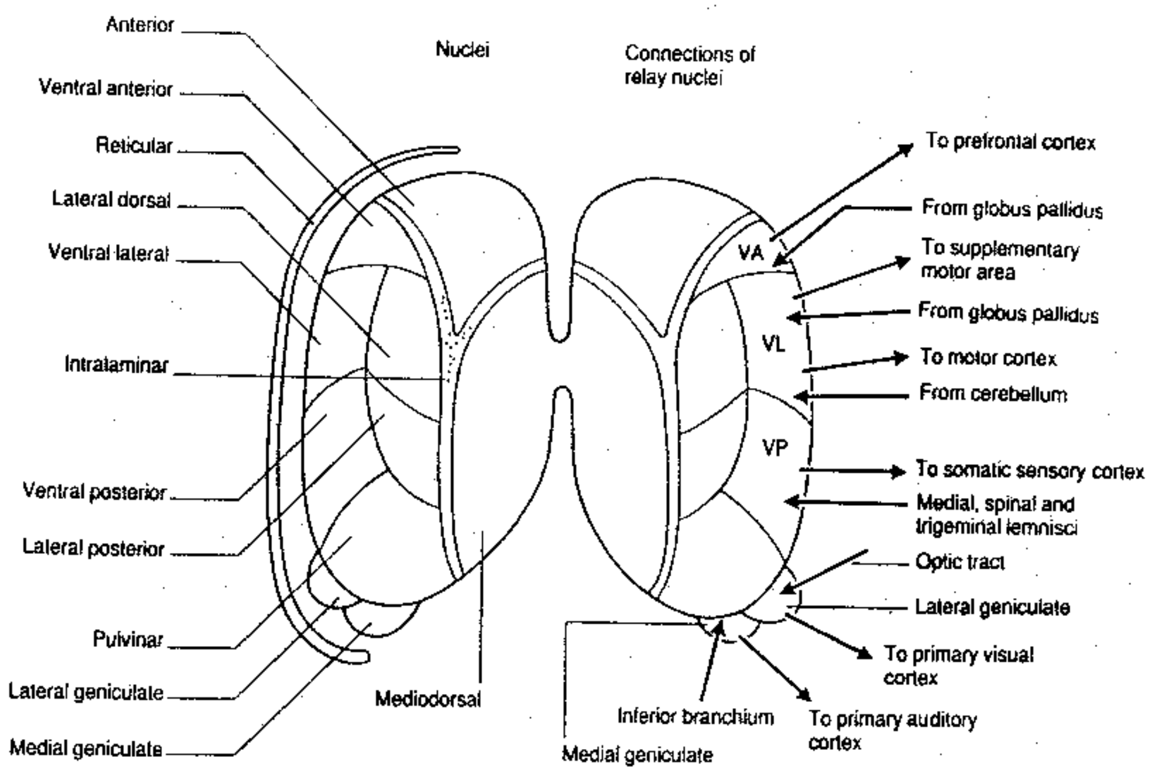


The LGN takes inputs from the **same side of each eye** (we have left side and right side of each eye RR,RL and LR,LL) where R=Right,L=Left. RR is paired with LR. Therefore, it has enough information to build stereoscopic VISUAL ALGEBRA 3D messages with some processing because it's the same image from each eye between the nose.

This is a really nice design for an Android to use. Makes complete sense.

Completing the Message Circuit back to the Thalamus and the SC

The messages which are routed to the Dorsal and Ventral system produce resulting messages of different types. One third of the brain handles vision requirements and according to this theory they end up going back in the Thalamus where the SC-RR is located and build what we think of as “Reality”.



Here is an example of the Thalamic Nuclei which maps to the major messaging centers in the brain. The "From Cerebellum" would handle the brain messaging which includes the VISUAL ALBEGRA.

The Virtual Self And The Mind's Eye

In the theory, the Virtual Self is placed inside the SC-RR. This is achieved by means of Room Location algebra.

SC ROOM LOCATE VIRTUAL SELF X,Y,Z

It's been shown that if one wears glasses showing a person in another location that the Virtual Self feels they are standing in that location.

If one takes off the glasses, one switches back to their current location.

The Visual Algebra helps determine the Virtual self location.

The Virtual Self is connected to all the SC Message Queues and therefore is the focal point of Conscious messaging.

SC LOCATE VIRTUAL SELF ROOM X ATTACH CONSCIOUS MESSAGING

An Android should use this idea and be able to switch from one room to another

Day And Night Messaging

The day and night messaging is controlled by VISUAL ALGEBRA which monitors the time of the day and the amount of visible light.

When it is time to sleep, the VIRTUAL SELF is placed in a sleep room and much of the conscious messaging is no longer routed to it and we “dream”.

SC LOCATE VIRTUAL SELF ROOM “SLEEP” DETATCH CONSCIOUS MESSAGING

Sleep should be a part of an Android and go into some kind of maintenance mode to organized thoughts and memories and do system checks on its body.

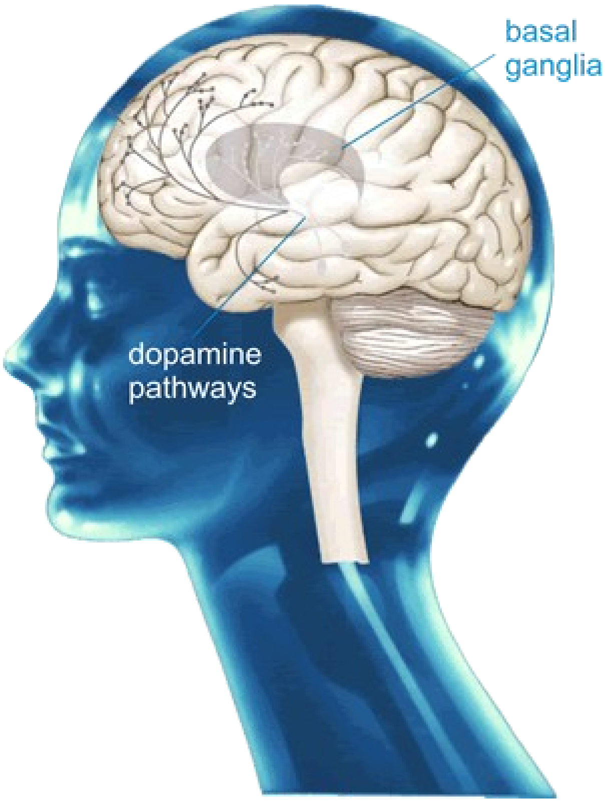
Chapter 9 Music And Speech

Here we will cover how music and speech affects the brain. To do this we need to understand how the brain handles motor function. Also music and speech affect us in different ways plus there are some conditions related to these sections of the brain which help us understand how these parts of the brain work.

Motor Function And The Basal Ganglia

There are three main circuits for motor function. The Basal Ganglia controls the motor function and in all three cases, the message types are routed through the Thalamus. This includes the motor control not just for movement but also for our voice which includes singing and speaking.

The main three components of the basal ganglia are the striatum (caudate nucleus and putamen), the globus pallidus (the substantia nigra and the nucleus accumbens) and the subthalamic nucleus. It is located close to the Thalamus.



Rehearsal Circuit

The rehearsal circuit is where we repeat a motor function like dancing and perfect it. Here we use the Putamen.

Putamen → thalamus → motor cortex   (message loop back to Putamen)

This is how we "practice" until we get the hang of something like new movements etc;

Reward Circuit

One we learn something and become good at it, the Nucleus Accumbens remembers and rewards us by making us feel good and so forth. Here we anticipate something which we know can fulfill a GOAL. I will talk about this later.

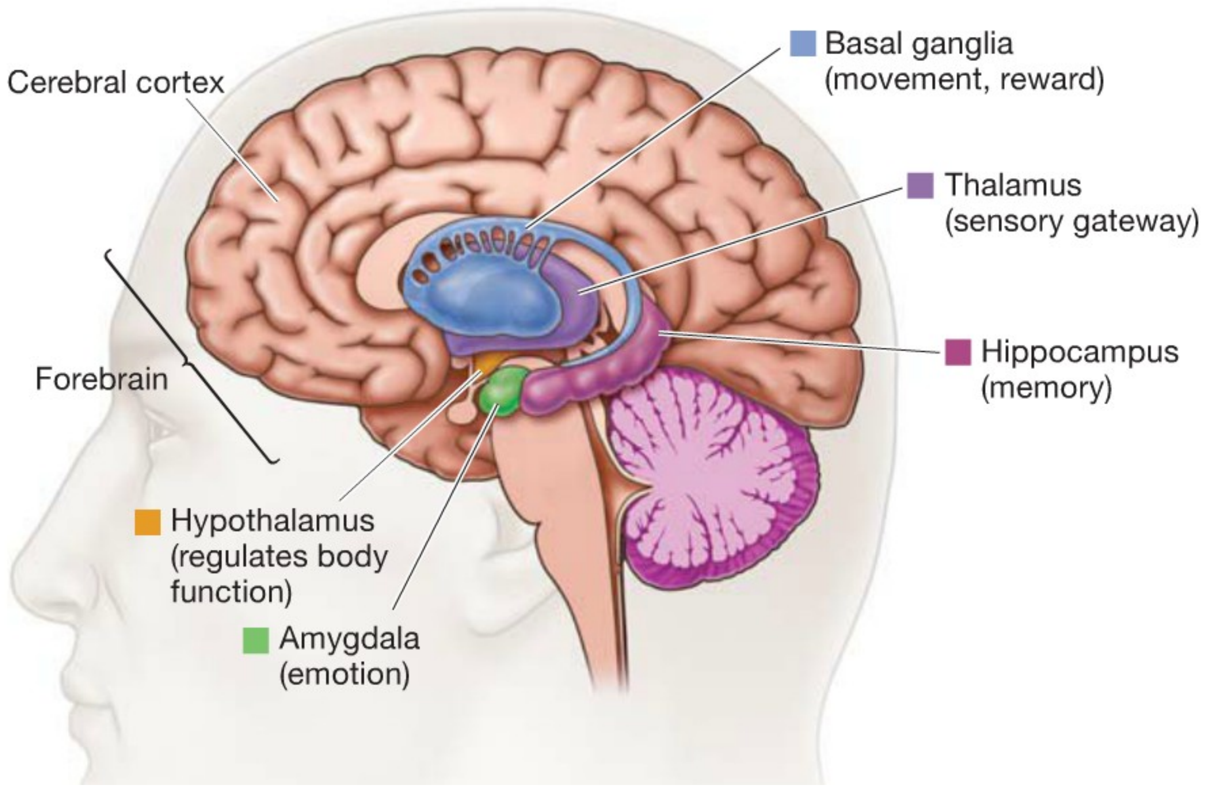
Caudate accumbens → Thalamus → cortical System (message loop back to Caudate accumbens)

Emotional Muscle Movement

Depending on our emotional state our face can look happy or sad. Our posture is also affected. We may slump or stand tall.

Limbic circuit → Hippocampus → thalamus → (message loop back to limbic)

All three join and go through the message centers in the Thalamus. Here they are combined in the overall message queues which combine to form part of our consciousness in the SC.


Cerebral cortex
🟦 Basal ganglia (movement, reward)
🟪 Thalamus (sensory gateway)
🟪 Hippocampus (memory)
Forebrain
🟧 Hypothalamus (regulates body function)
🟩 Amygdala (emotion)

An Android should therefore also have

Learning Circuit related to motor control (stores them once they are learned)

Building MOTOR MESSAGE STORIES

Reward Circuit (doing a good job for a learned set of movements)

Predicting MOTOR MESSAGE STORIES from stored message stories

Emotional Circuit (happy or sad depending on how goals are reached)

GOAL RELATED EMOTIONAL ALGEBRA which produces MOTOR MESSAGE stories

Rhythm And Pairing Sound With Motion

One of the most enjoyable things we can do as people is to listen to music and move to it. Music has different properties. One of these is Rhythm. In this theory, the two hemispheres of our brain need to co-ordinate to do motor functions.

For example to walk, we need to place one foot in front of the other.

MOVE LEAD LEG FORWARD

NOTIFY OTHER HEMISPHERE WHEN COMPLETELY DONE

MOVE OTHER LEG FORWARD

NOTIFY OTHER HEMISPHERE

and so on REPEAT

This requires timing.

In the brain according to this theory we have pairing of Sound and Vision.

PAIR(SOUND,VISION) and SYNCH across HEMISPHERES

Typically for vision we need to keep both our eyes focused on the same point and upper body movement control of our neck and so on.

However, we also have pairing of Sound and Movement

PAIR(SOUND,MOVEMENT) and SYNCH across HEMISPHERES

Therefore to walk we have rhythm which is message syncing across hemispheres and related to timing and movement but **also sound**.

In this theory, this is the basis of Rhythm namely the pairing of Sound and Movement of ones body.

Therefore sounds can make us move in different ways which I'll cover shortly. So even though we may walk without sound, in our mind there are related sounds to walking like this because of the pairing.

An Android should also implement this approach and have rhythm which will be essential for smooth movement and auditory processing.

The Metronome And Hemisphere Synchronization

In music the Metronome is used to synchronize with the beat of a tune.

According to this theory because Sound and Movement are paired, the Metronome reflects the way that the hemisphere inter-communication occurs, from one hemisphere to the other.

The human ear will pick out one of the metronome beats to be the major one. The reason for this is that motor function which is paired with sound needs to have a starting point. We call this the dominant hemisphere.

Typically we might be right or left handed which means that GOAL motor messaging starts on this side of the brain which maps to a certain limb.

When a newborn starts to crawl, they move a hand on one side of the body and then the limb on the other side of the body. Therefore, like a metronome, the Motor Messages switch from one side of the brain to the other.

The same is true for walking. We choose a lead foot.

Also for vision we have a lead eye, then the other one follows.

MOTOR MESSAGING

DO MOVEMENT HEMISPHERE A

DO MOVEMENT HEMISPHERE B

… etc;

Because sound is related to this process, the types of sound that we hear can match various types of motion we experience which I will explain shortly.

Tempo And Messages Per Minute

In music the tempo of music is broken into beats per minute. A metronome counts. A bar of music has so many beats. In this theory, this maps to the number of Message Stories transferring from one hemisphere to the other per second. One beat represents the pairing sound of Sound and Movement. Typically when people hear a sound they might tap a foot or a hand or nod their head to a beat. According to this theory, this muscle movement is an early sign that Sound has an associated Movement message pair and leads into the whole developed area of dancing to music. As the tempo increases of the music, this is used in placed like gyms where people work out because it aids in the use of muscle movement. The faster the tempo, the more muscle movement signals that are sent across the hemispheres. Typically a sheet of music might show beats per minute at one twenty which is two per second.

How one chooses to move to sound appears to have defaults like foot tap but can be overridden by the individual.

Also it's been shown that one cannot dance to flashing lights on their own. This is because there is not a pairing of muscle movement with VISUAL ALGEBRA according to this theory.

An Android should also support pairing of sound with limb movement across a two hemisphere model.

Polyrhythmic Beat And Counting

There can be more than one beat rhythm per minute. In music one can have 3 over 2 or 2 over 3 or 4 over 3. Typically, a common base is found where under underlying Metronome beat is setup with a base 6 for example.

It's up to the musician to count the beats and then apply the movements on one side or another. This maps to a base beat.

Some complex piano pieces or drum routines require different beats for different hands and feet. To do this, these accomplished artists need to do some kind of counting while playing or have the assistance of a metronome or other cue.

COUNT 6 BEATS

APPLY MOTOR MOVEMENTS EVERY 2 BEATS ON HEMISPHERE A

APPLY MOTOR MOVEMENTS EVERY 3 BEATS ON HEMISPHERE B

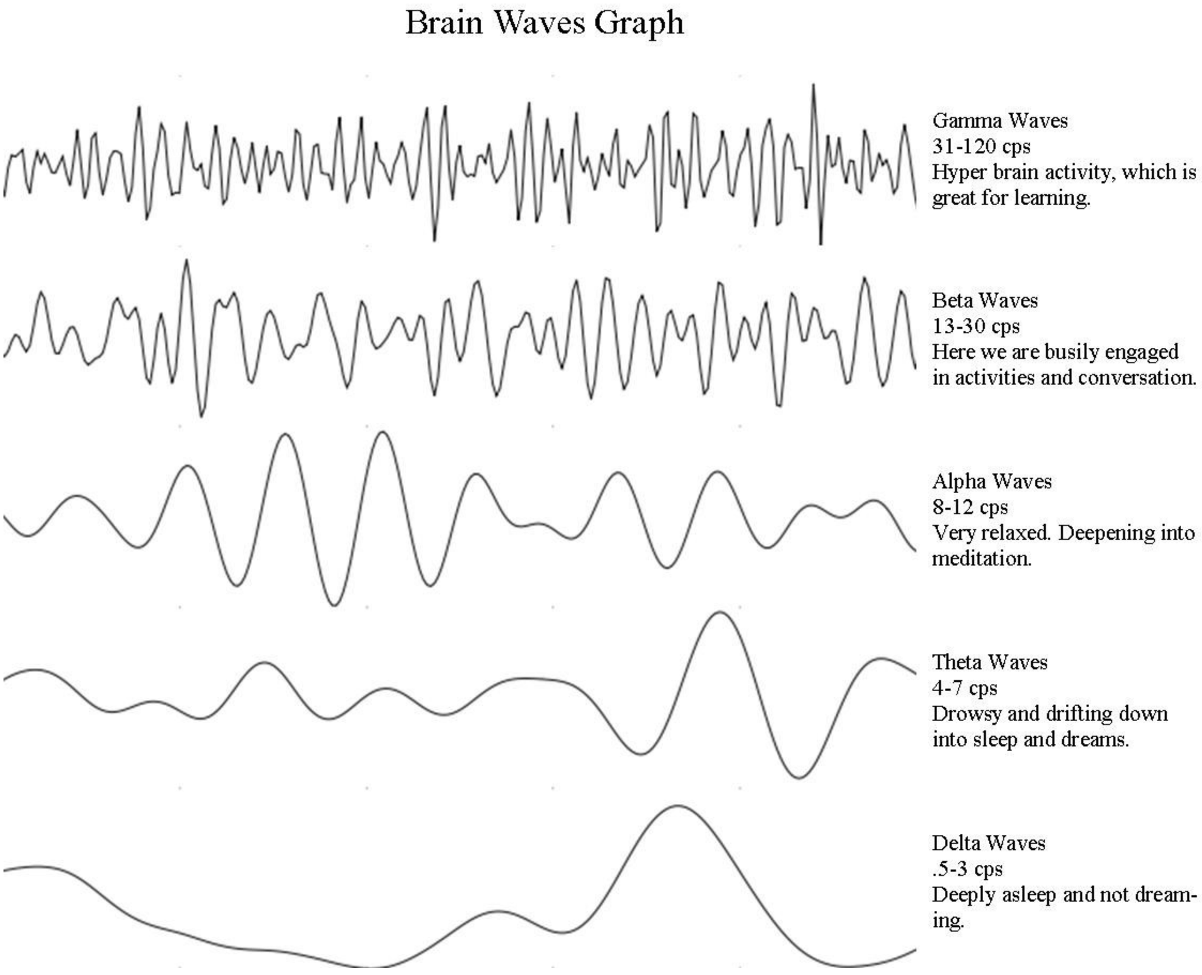
In a more complex example there are many layered beats off a base beat to synchronize the hemispheres. This is analogous to how the brain works where co-ordination is required between the hemispheres.

An Android should therefore have a Base Beat between the hemispheres which co-ordinates one hemisphere with another for general activities. Motor movements should be defined in terms of their Hemisphere count. This is a good example of using a logical AND for all the motor movements on each hemisphere.

A very musical brain therefore has highly developed sense of timing between hemispheres and what to do and when. Accomplished dancers will also have this ability.

Brain Waves, Message Type And Message Passing Rate

Traditional Neuroscience measures brain wave activity using electrodes on the scalp and so on. Research has shown that there are different states for the mind in terms of these brain waves.


Brain Waves Graph

The unit of measure is in Hertz which is waves per second.

In this theory what these waves represent are messages being passed from one Neuron to another. Therefore these graphs represent the **rate of message passing between brain cells** depending on ones state (awake, contemplating, active, asleep etc;). The height (amplitude) of the wave can be mapped to different message types.

If one adopts this viewpoint on can see that at the peak we process over twenty messages per second. These messages are aggregated on message queue brain fibers and then sent to the Thalamus.

When we are relaxed and being creative we have 8 to 12 messages per second. Therefore one sees that if the Thalamus is not too busy one tends to be more creative and have ideas.

When one is asleep the Conscious messages drop off and the Thalamus goes into another state according to this theory and the messages rate is almost zero. Here our virtual self is switched into another room and we mostly look up memories and dream.

Studies have shown that when one is stressed one may have 60 thoughts per second which maps to message stories so it looks like the brain SC is comfortable with less than this.

An Android should adopt limits like this and have some kind of monitoring of the number of messages going through the SC Queues.

Music And Emotion

Music has always been associated with emotion. By implication sounds in general are associated with emotion for example when one is very sad one may cry out loudly. Also if one is angry one may become louder.

Therefore **sound and emotion are paired.**

PAIR(SOUND,EMOTION) and SYNCH across HEMISPHERES

It's been shown by studies that the emotional coupling is not directly coupled in the same way for everyone. For example, certain composers or songs may elicit different strong emotional responses depending on who the person listening is.

Strong emotional responses according to this theory are related to underlying GOAL settings.

GOAL A SUCCEED

versus

GOAL A FAIL

produce strong emotional responses. Large sporting occasions are a good example of this when in a final only one team is the winner while the other loses. The brain highly rewards success versus the body produced stress hormones of failure.

Liking and disliking are part of the SOCIAL NETWORKING ALGEBRA where one has likes and dislikes.

GOAL LISTEN TO MUSIC TYPE A (LIKE)

versus

GOAL DO NOT LISTEN TO MUSIC TYPE A (DISLIKE)

If one listens to it and has it as a goal, then one feels good and gets an emotional response because a goal is fulfilled.

Imagination, Rehearsal And The Virtual Self

Where does the brain locate the Virtual Self and the SC GC in this theory? This is a place where we can imagine things and we place ourselves in it. According to this theory the Thalamus is the Message Center for the SC. The GC rooms where we are placed are in the adjoining regions of the Basal Ganglia, in particular the putamen.

Research has shown

-**Imagined** Vs. Muscle Movement: Head of the right **caudate nucleus** and anterior part of the left **putamen** in the basal ganglia.

-Muscle Movement Vs. imagined: Anterior part of the right putamen, posterior part of the putamen and ventrolateral nucleus of the thalamus bilaterally.

So from this we can see that the putamen on one side handles imagined movement for rehearsal and the other putamen handles the muscle movement portion. The caudate nucleus is responsible for cognition and co-ordination which according to this theory, one imagines a task.

Therefore one imagines ones self in a SC GC Room which is created in this part of the brain.

In an Android therefore one should have a SC GC Room which connected to the SC Messages and also has a mapping to the Motor Corex where we send out Movement Messages. It appears that there are two copies of this room in the brain. In one, we can imagine or project future movements which is mental projection of movement. In another, we then can map those movements to actual muscle movements and the two tasks are separate.

It makes sense that an Android could operate like this.

So one is a

SC Planning Room

SC Muscle Movement Room

Message Algebra And Moving Ones Hand

Research has shown that hand movements involve certain messages going to all fingers to do a STOP and another to do a MOVE in order to do a finger movement for example. This is shown as an example of evolution in action. In this theory, this shows how the Message Algebra works where messages are combined.

AND [

FREEZE ALL FINGERS

MOVE THUMB

]

Therefore this is an example of Message Algebra in action. By having atomic operations one can control certain parts of a body while providing action for just one.

An Android should support these types of operations to build message stories.

Huntington's Disease

When someone has Huntington's disease a person suffers from involuntary muscle movements called Chorea and their cognition is impaired. The region of the brain is the Nucleus Accumbens. According to this theory, this is where the Virtual Self is located and where imagined movements are planned by the brain.

Therefore if this part of the brain degenerates according to this theory, one will produce imagined movements which make no real sense and they will be passed over the motor function area. Also, the Virtual Self will be affected by this and find it increasingly hard to make sense of the SC GC room as there need to be Message Story look ups to do future planning such as cleaning ones face and so on. The idea of this part of the brain according to this theory is to figure out how to do the next things in ones daily chores so without the ability to do this, it can be very frustrating for a sufferer. From limited research, it also looks here like additional proteins are causing the neurons to age too rapidly. I have no proof of this idea. This is pure theory. Hopefully a cure can be found soon either way.

Visualization And Cognition

The nucleus accumbens can therefore be seen as a Visualization center for the Virtual Self. The ability to set a personal goal and then imagine the next steps to achieve this goal is the basis of human intelligence in my opinion.

The simple steps are to visualize the goal, figure out each step and then implement each step. To do this the brain needs to have some kind of "future imagining"center and this part of the brain appears to serve that function.

This is pretty crucial to Android Intelligence.

It should be able be given a future goal and then figure out the next steps and do them or at the very least be told what to do.

The ability to take a copy of the SC RR and push it forward into different states without actually doing it is how we will regard an Android as "intelligent" in my opinion. The basis of cognition involves thoughts (messages), goal and problem solving using an SC RR.

A simple example is a board game where one can imagine the next move on the board without actually doing it.

So for an Android we have

SC GC RR (Real time reality room)

SC GC PPR (Predict Plan Room) – A "Future"

The "Predict Room" is where we imagine how we might implement a solution without actually doing it which is the basis of human planning. Because the room is based on a message algebra we can apply test Message Stories to these rooms and try to figure out the outcome.

This is a pretty key android feature.

Tourettes, Drumming And Deep Brain Stimulation

Tourettes involves involuntary muscle spasms which can be audible. According to this theory, messages are generated in the brain which cause the body to be hard to control. However, some sufferers have found that music in particular drumming overrides the symptoms. Also, the medical technique of deep brain stimulation (placing electrodes near the basal ganglia) can lessen the effects of Tourettes. According to this theory, the reason for this is that both Drumming and Deep Brain stimulation of the Basal ganglia causes higher priority messages to override Tourettes messaging. Drumming requires synchronization between the hemispheres and causes the other messages to be overridden. This is similar to Acupuncture where certain locations in the body override pain messaging.

In an Android therefore, we should have a message hierarchy.

This is analogous to message algebra.

DRUMMING MOVEMENT PRIO 0

INVOLUNTARY MOVEMENT PRIO 1

Prio 0 is implemented over Prio 1.

Movement to music is also very old and is a higher priority message.

Some doctors have also witnessed patients who have been in a virtual coma get up and sing and dance to music even though they do not react to normal stimuli. According to this theory, this is because music and singing deals with very old messaging with high priorities that can override an underlying cerebral cortex condition which incapacitates the patient.

Visual Stimulation And Emotional Algebra

Many popular shows involve Celebrities have to be trapped with insects or eat insects. This visual algebra creates a strong emotional response to the viewer. Therefore although Vision is not related to movement and rhythm, a picture with a strong emotional response such as eating an insect can cause one to react strongly.

Evolutionary it's programmed into us to not eat certain insects and to watch someone doing it or being trapped with insects or reptiles creates a strong response.

```
VISUAL ALGEBRA EAT INSECT
LOOKUP RELATED FOOD MEMORIES
MESSAGE STORY RESPONSE[
EMOTIONAL ALGEBRA DISGUST
MOTOR FUNCTION REACTION
]
```

An Android should have rules on what it should and should not do. For example, it should keep clear of any location and behavior that is harmful to it.

PTSD And The Emotional Threshold

Many people who experience very traumatic experiences record these in their hippocampus and store memories with messaging which is very high in EMOTIONAL ALGEBRA. On occasion these memories surface and conflict strongly with the basic goals of the person such as conflicting with their SAFETY and SECURITY goals. This produces an alarm in their Amygdala which in turn produces a stress response and it can cause a continuous feedback loop where the body becomes overwhelmed.

Typically when a person is giving birth or in an accident, the body does not record each moment for a reason, it's too stressful for the body to remember. Therefore memories with an experience that is too traumatic needs to be dealt with in some kind of counseling or other approach to lower the emotional impact. However this is easier said than done and may require additional treatment.

An Android should have some kind of EMOTIONAL ALGEBRA threshold for message stories. Typically some people have a higher threshold than others and appears to vary from one person to the other.

Music And Motor Function

According to this theory music is mirroring the messaging patterns in the brain. The Do Re Mi Fa So La Ti Do scale represent a muscle movement from one place to another. For example, one might typically move ones arm gently up or along an imaginary key board. One might say music is 'funky' which translates into a movement of one's body in certain ways. Rock'n'roll maps to certain dance styles and so on.

Chords are an interesting case because they deal with more than one sound which sound 'good'. This was discovered by trial and error. According to this theory this is because the brain likes multiple message streams and they match ones one would normally get in the brain.

Rhythm can indicate chasing or being chased. A pace maker in a race is how many other runners determine the overall speed of the race. This would be the main beat in a song. Music before the main beat is chasing as in a race and after the main beat is being chased by.

An Android should understand what constitutes good music.

Music And Grammar

The grammar of music like language represents the appropriate or pleasant sounding next parts of music. This can be a song one already knows or sounds which sound good because the brain likes the note patterns. If notes are off and slightly unexpected this can represent and be interpreted as a song with emotion where a singer might sound unhappy or distressed. In other words the predicted next correct notes are not there and there is distress. Composers who understand music know how to do this. Additionally music which sounds good has the expected result and can change mood positively.

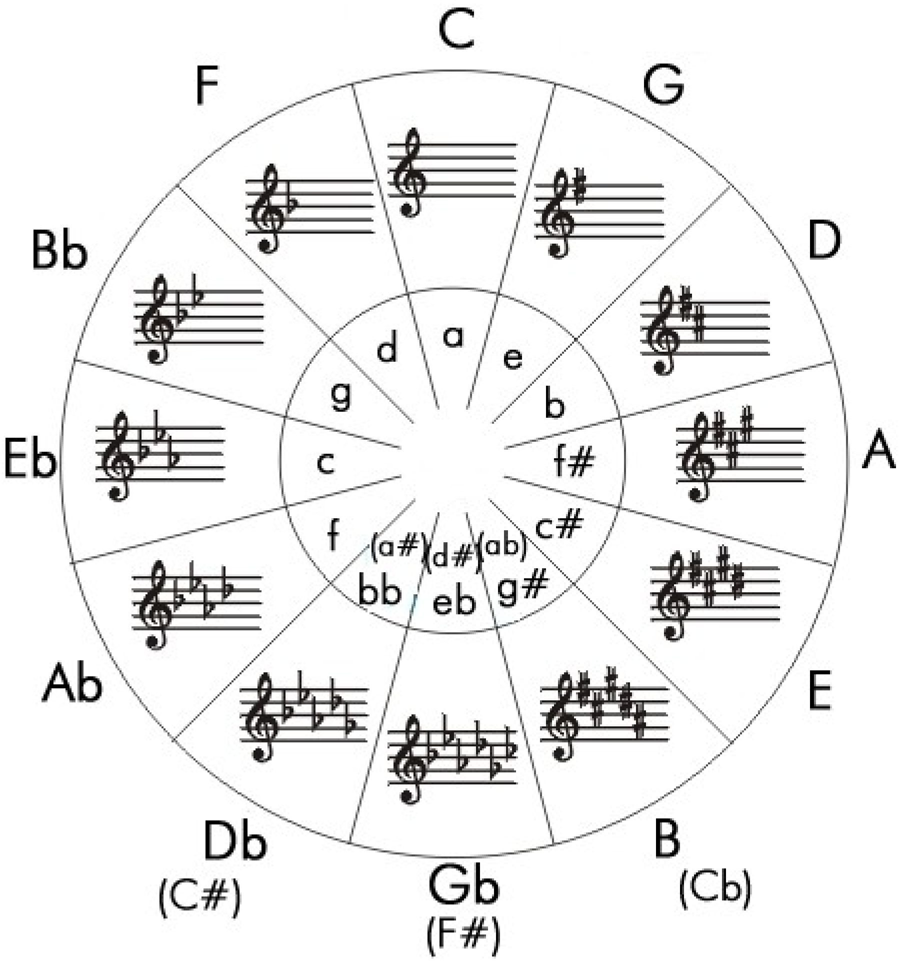
The sound of a talented singer's voice can also invoke emotion. Many operas use this to great effect. In moments of sadness people can sound distressed which maps to high frequency which maps to many messages of high emotional content passing through the SC.

The frequency of messages going through the SC maps to a higher pitched sound according to this frequency which maps to higher motor movement or emotion.

An Android should understand the grammar of music. This is a pretty big topic.

Circle of Fifths And Message Rate and Speed

Music has sharps and flats which match to the notes between the major ones. The Circle or Cycle of the Fifths indicates the notes which sound correct to us. The number of flats and sharps increases as one changes key incrementally to keep it sounding correct. Why is this? The answer according to this theory is to do with the number of messages which go through the SC and match how our motor function gauges an increase or decrease in motor function messages.



An Android should be aware of this and try to co-ordinate motor function messages with music notes and the Circle of Fifths. This way music notes will also sound good to an Android in the same way they sound good to a person.

16 Different Personality Types and Myers Briggs

When people play music they do not always play the music the same way. Humans have different personality types which means they approach things in different ways.

What this means in an Android is that there are different goals in the SC which determine its SC message configuration and overall focus.

In an Android we'll give each competing goal a percentage totaling one hundred percent. The idea is that one is not completely extravert or introvert for example.

This leads us onto the idea of Android types where they do different types of tasks. A protocol droid for example needs to like to talk to people.

Each derived goal in the table generates related messages which end up in the SC.

Function	
Introvert versus Extrovert	• Extraverts are "action" oriented, while introverts are "thought" oriented. • Extraverts seek "breadth" of knowledge and influence, while introverts seek "depth" of knowledge and influence. • Extraverts often prefer more "frequent" interaction, while introverts prefer more "substantial" interaction. • Extraverts recharge and get their energy from spending time with people, while introverts recharge and get their energy from spending time alone; they consume their energy through the opposite process. Extravert (%) GOAL DO ACTION GOAL BREADTH OF KNOWLEDGE GOAL FREQUENT INTERACTION

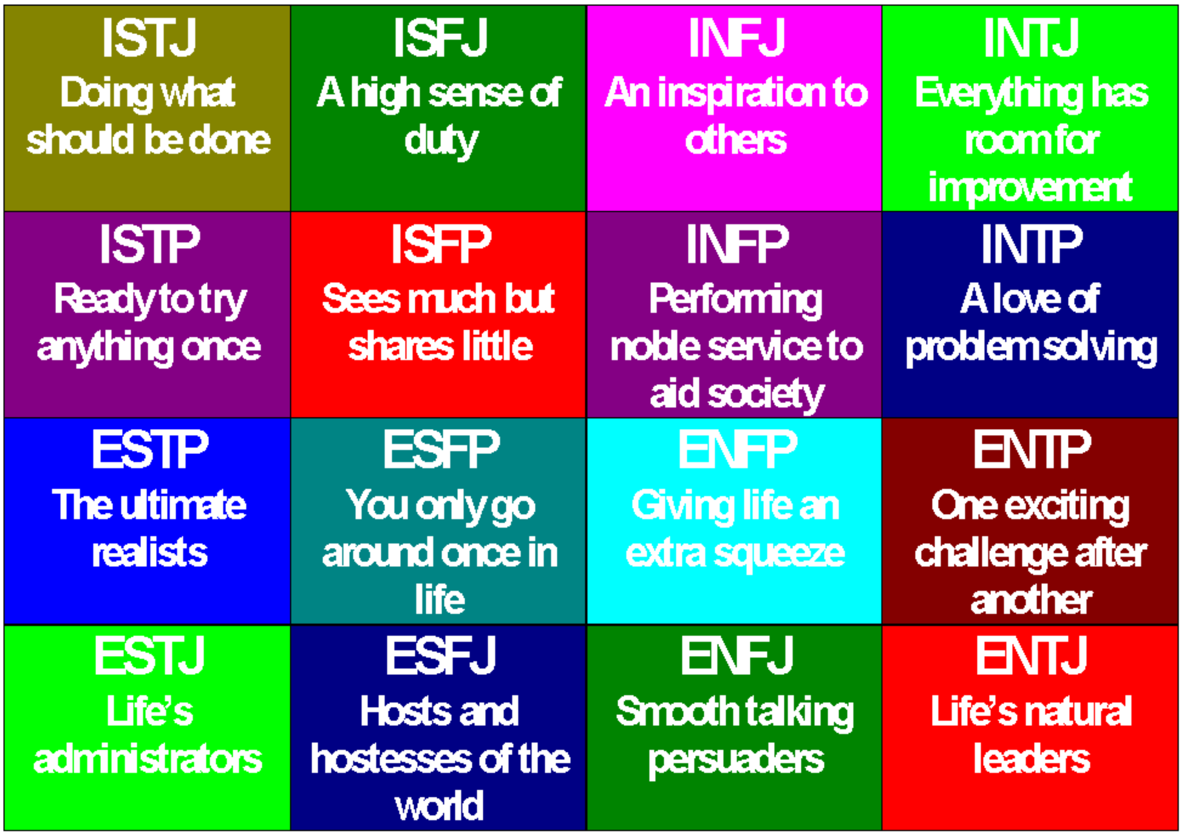
	GOAL SPEND TIME WITH PEOPLE Introvert (%) GOAL THINK ABOUT THINGS SC PPR GOAL SUBSTANTIAL/DEPTH INTERACTION GOAL SPEND TIME ALONE
Sensing versus Intuition	*Sensing* and *intuition* are the information-gathering (perceiving) functions. They describe how new information is understood and interpreted. Individuals who prefer *sensing* are more likely to trust information that is in the present, tangible, and concrete: that is, information that can be understood by the five senses. They tend to distrust hunches, which seem to come "out of nowhere". They prefer to look for details and facts. For them, the meaning is in the data. On the other hand, those who prefer *intuition* tend to trust information that is more abstract or theoretical, that can be associated with other information (either remembered or discovered by seeking a wider context or pattern). They may be more interested in future possibilities. For them, the meaning is in the underlying theory and principles which are manifested in the data. Sensing (%) GOAL DETAILS AND FACTS HIPPOCAMPUS

	GOAL THEORIES BASED ON DATA Intuition(%) GOAL DETAILS BASED ON SC PPR IDEAS GOAL LOOSE DATA COUPLING HIPPOCAMPUS GOAL FUTURE POSSIBILITES
Thinking versus Feeling	*Thinking* and *feeling* are the decision-making (judging) functions. The thinking and feeling functions are both used to make rational decisions, based on the data received from their information-gathering functions (sensing or intuition). Those who prefer *thinking* tend to decide things from a more detached standpoint, measuring the decision by what seems reasonable, logical, causal, consistent, and matching a given set of rules. Those who prefer *feeling* tend to come to decisions by associating or empathizing with the situation, looking at it 'from the inside' and weighing the situation to achieve, on balance, the greatest harmony, consensus and fit, considering the needs of the people involved. Thinkers usually have trouble interacting with people who are inconsistent or illogical, and tend to give very direct feedback to others. They are concerned with the truth and view it as more important. Thinking (%)

	GOAL RULE BASED HIPPOCAMPUS GOAL INFORMATION GATHERING GOAL DIRECT COMMUNICATION GOAL FACTS Feeling (%) GOAL EMPATHY GOAL HARMONY GOAL CONSENSUS
Judging versus Perception	Myers and Briggs held that types with a preference for *judging* show the world their preferred judging function (thinking or feeling). So TJ types tend to appear to the world as logical, and FJ types as empathetic. According to Myers,judging types like to "have matters settled". Those types who prefer *perception* show the world their preferred perceiving function (sensing or intuition). So SP types tend to appear to the world as concrete and NP types as abstract. According to Myers, perceptive types prefer to "keep decisions open". Judging (%) GOAL LOGICAL GOAL MATTERS SETTLED GOAL CONCRETE WORLD

	Perception (%) GOAL KEEP DECISIONS OPEN GOAL EMPATHIC

In Myers Briggs this leads to a table of people types.



ISTJ Doing what should be done	ISFJ A high sense of duty	INFJ An inspiration to others	INTJ Everything has room for improvement
ISTP Ready to try anything once	ISFP Sees much but shares little	INFP Performing noble service to aid society	INTP A love of problem solving
ESTP The ultimate realists	ESFP You only go around once in life	ENFP Giving life an extra squeeze	ENTP One exciting challenge after another
ESTJ Life's administrators	ESFJ Hosts and hostesses of the world	ENFJ Smooth talking persuaders	ENTJ Life's natural leaders

In this theory one is not just purely ISTJ, one can be a percentage of each type which generates a percentage of messages types depending on the moment in time and some circumstances.

Also one should take into account the fact that there are two hemispheres. It's been shown in studies that one hemisphere can have a different belief system and set of values that the other in some interesting cases.

Therefore, the table is a good guide but in the theory the Android will have a more flexible configuration based on percentages. Clearly if you want one type of android then use 100% in one category over another.

Note also that having a goal like GOAL LOGICAL requires some kind of a **Goal Manager App** which understands what this means in terms of the Android. This will be covered in a Logic and Reason Chapter. For now it's enough to just have the personality types and the percentage balances and high level goal definitions.

Chapter 10 Sensors And Self Awareness

In this chapter, we touch upon the idea of how the Virtual self understands the outside world via its sensors in the skin and so on and what part of the brain deals with this. This is a pretty important topic. In the future, one could imagine an Android sent to a world that would be hostile for a human being in terms of temperature and pressure but could be survivable for an Android. The quality of the body sensors also help add to the sense of reality for the Virtual Self in the SC Rooms.

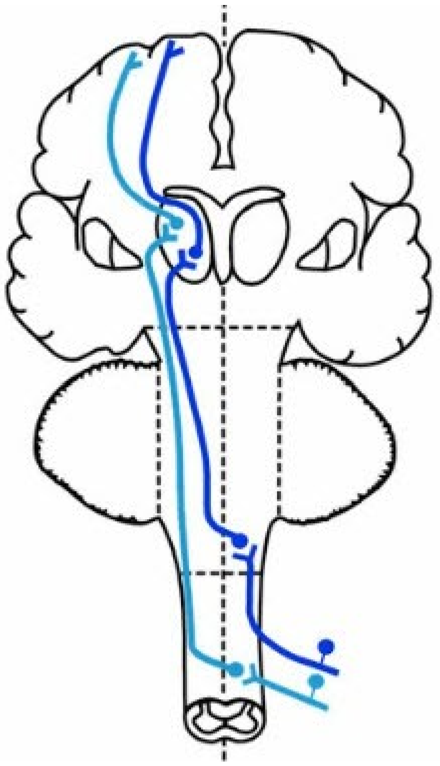
Two Sensor Circuits Sharing Similar Pathways

There are two circuits. Each passes through the Thalamus LVN and then onto to Somatosensory area of the brain.

The pathway to the Thalamus is slightly different for certain sensations but both end up in the same place roughly and are then routed onto the Somatosensory areas. One group pass through the Dorsal part of the brain which related to our position of self both virtual and physical. The other does not involve the Dorsal part.

Pain and Temperature do not involve the Dorsal system and therefore the “feeling” of pain and temperature is generated by the Virtual Self. One must react to it in an appropriate manner but if one moves a limb or has ones hair tousled by the wind, that has actually happened to the physical self.

Fine touch, proprioception, wind moving over skin, vibration detection, limb positions etc; are covered by the Dorsal circuit and involves updating the physical self as well as the virtual self.



For an Android it makes sense to also break it up this way.

We need to define SENSOR ALGEBRA covering

PAIN

VIBRATION

PROPRIOCEPTION

TEMPERATURE

SKIN STRETCH

SKIN PRESS

FINE TOUCH

We need to pair these messages with BODY LOCATION MESSAGES

So

PAIR(SENSOR ALBEGRA,BODY LOCATION)

e.g.

TEMPERATURE, ARMS

For example on a sunny day. These messages go into the Thalamus and then onto the Sensor Center proper.

Body Location versus Body Position

We need two types of Algebra for any body. We need BODY LOCATION algebra which is the means by which a being indicates where a message is coming from in a body. Even if a BODY position has changed, the location remains the same.

We also need a BODY POSTION algebra which indicates the arrangement of the body. For example, one can image a person doing a certain work task and moving their limbs. Sitting down requires the body position of certain limbs. This type of messaging needs to be shared with the Virtual Self and the Physical Self. In some life forms this type of information can be quite complex like an octopus. However if the shape is a car with some kind of intelligent system within it, the position might indicate if windows are down or doors are open or closed.

Body position for person

ELBOW 'N' DEGREES

FINGERS EXTENDED

For an octopus

LIMB 'A' SHAPE 'B'

LIMB 'C' SHAPE 'D'

For an Automobile

PASSENGER WINDOW FULLY OPEN

DOORS CLOSED

FRONT LIGHTS UP

Virtual Self and Physical Self And Dedicated Circuits

The Virtual Self and the Physical Self share sensory information. Some things like pain and temperature have to be mapped to some kind of experience for the Virtual Self as I have shown.

In terms of processing, it's quite computationally expensive to maintain a Virtual Self and a Physical Self and therefore it makes sense as it is done in the body to have separate circuits for the two operations but they should share the core messaging.

However both to need to be synched and this is where the Thalamus Messaging center serves the purpose of coordinating the two circuits. As a result, an Android should have separate dedicated circuits like this as well.

This requires that the Virtual Self and the Physical Self messaging are paired.

Core And External Temperature And The Metabolic Rate

In the human body, skin sensors detect outside temperature. They operate in a range from normal getting hotter and from that range getting colder. Therefore they create a skin temperature range of hot and cold. Additionally, if the temperature gets too hot or too cold like frost bit for example, then a different type of message is sent to the brain to indicate skin damaged.

SKIN TEMPERATURE AVERAGE NORMAL A

SKIN TEMPERATURE SAFE RANGE PLUS B

SKIN TEMPERATURE SAFE RANGE MINUS C

SKIN TEMPERATURE DAMAGE PLUS D

SKIN TEMPERATURE DAMAGE MINUS E

The internal core temperature is mapped by the hypothalamus. This needs to be maintained around a fixed average human temperature. Internal organs need regulated temperatures for normal operation of the body.

CORE TEMPERATURE AVERAGE NORMAL A

CORE TEMPERATURE SAFE RANGE PLUS B

CORE TEMPERATURE SAFE RANGE MINUS C

CORE TEMPERATURE DAMAGE PLUS D

CORE TEMPERATURE DAMAGE MINUS E

In the case of an Android it makes sense to have skin and core temperature as well.

If one under heats one can burn certain fats cells or one shivers to create muscle movement which generates heat. Also body hair raises and protects the skin from external cold or one can just 'wrap up' by putting on something to keep ones heat in place.

If one overheats one can adjust ones clothing style and the body will also use sweat glands to release water which adsorbs heat and cools down the body. Alternatively, one can enter a building with air conditioning. These are examples of different strategies to maintain the correct body temperature.

OVERHEAT STRATEGY A

UNDERHEAT STRATEGY B

An Android should have overheat and under heat strategies.   The temperature range of an Android or any device will be dependent on its shielding and the range of temperature of the objects that it contains. In a home environment for example, a computer typically uses fans to keep the CPU cooled.

The amount of work that a device does also is reflective of a human metabolic rate. So if an operation requires large graphical processing a GPU might heat up, similar to the way human muscle heats up the body of a runner. Therefore the overall metabolic rate of the Android needs to also be monitored.

METABOLIC RATE DEVICE A\

METABOLIC RATE DEVICE B

This needs to be mapped to the Core temperature of the Android.

Temperature And Pressure

In Physics temperature and pressure are closely related. Astronauts uses Space Suits to regulate temperature and pressure amongst other things but these are core components. If you increase the pressure for example on a planet landing on a foreign world or you are deep underwater, one needs to have core pressure maintained even if external pressure is changing as well. This is the same idea as temperature.

SKIN PRESSURE AVERAGE NORMAL A

SKIN PRESSURE SAFE RANGE PLUS B

SKIN PRESSURE SAFE RANGE MINUS C

SKIN PRESSURE DAMAGE PLUS D

SKIN PRESSURE DAMAGE MINUS E

The internal core pressure is mapped by areas like the inner ear and pockets of space in the body. This needs to be maintained around a pressure of ground level.

CORE PRESSURE AVERAGE NORMAL A

CORE PRESSURE SAFE RANGE PLUS B

CORE PRESSURE SAFE RANGE MINUS C

CORE PRESSURE DAMAGE PLUS D

CORE PRESSURE DAMAGE MINUS E

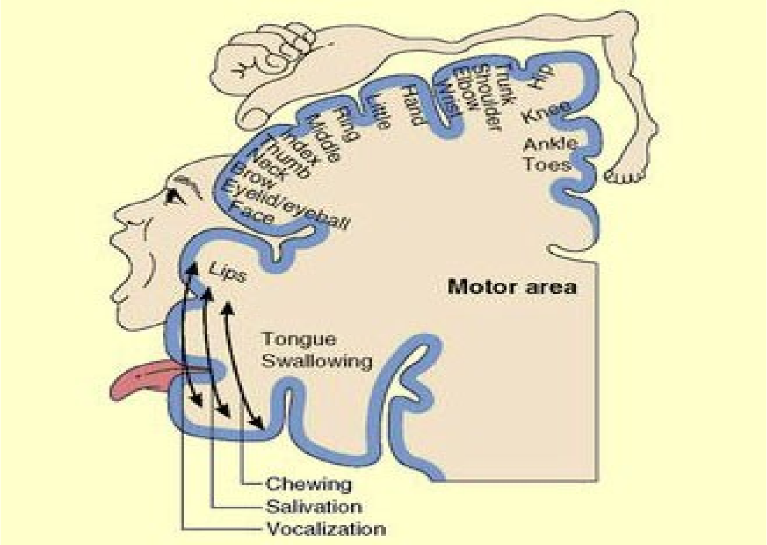
OVERPRESSURE STRATEGY A

UNDERPRESSURE STRATEGY B

Submarines use different strategies for these cases and examples of how the do this can be a useful practical example of how to implement the Message Stories. I leave it up to the interested student to apply this.

Mapping Body Location To Nerve Pathways And The Somatosensor Region

According to this theory, the virtual image of the body is stored in the Thalamus where the Self lives inside SC GC Room(s). This virtual images of ourselves needs to be able to sense and feel the outside world via various nerve endings. The Thalamus maps these nerve ending to a map of the body sensor points. Some parts of the body have more nerve endings than others.



Cortical homunculus

According to this theory, this is not the Virtual Body, these are the nerve endings of the Virtual Body which is housed in the Thalamus and adjoining structures. This part of the brain sends messages to do with POSITION sensation of the physical parts of the body. If stimulated with an electrode, we will get a feeling at the point where the mapping relates to.

The density of nerve ending varies as can be seen from the diagram. The face has more nerve endings that other parts for example.

Each part of the brain can therefore be seen as specialized message centers that are dedicated to different tasks, sending to and received from the Thalamus SC. Stimulating them will cause their messages to be activated and a "real" feeling will be experienced by the subject even though it may be in fact electrode related.

An Android should have a mapping like this as well.

Mapping Body Movement To The Somatomotor Region

The body's movement is also turned into messaging in the Somatomotor region (Primary Motor Cortex) which also relates to the movements of the physical limbs of and body parts.

This is a two way system where the Virtual Self and the Physical Self are paired with this type of messaging. As I have described earlier.

An Android should therefore have specialized Message areas for the body position and location.

Other Sensors

The body has sensors for other function like Acceleration and so on, these should also be paired within the SC.

Chapter 11 Memory, Mental Illness And Sleep

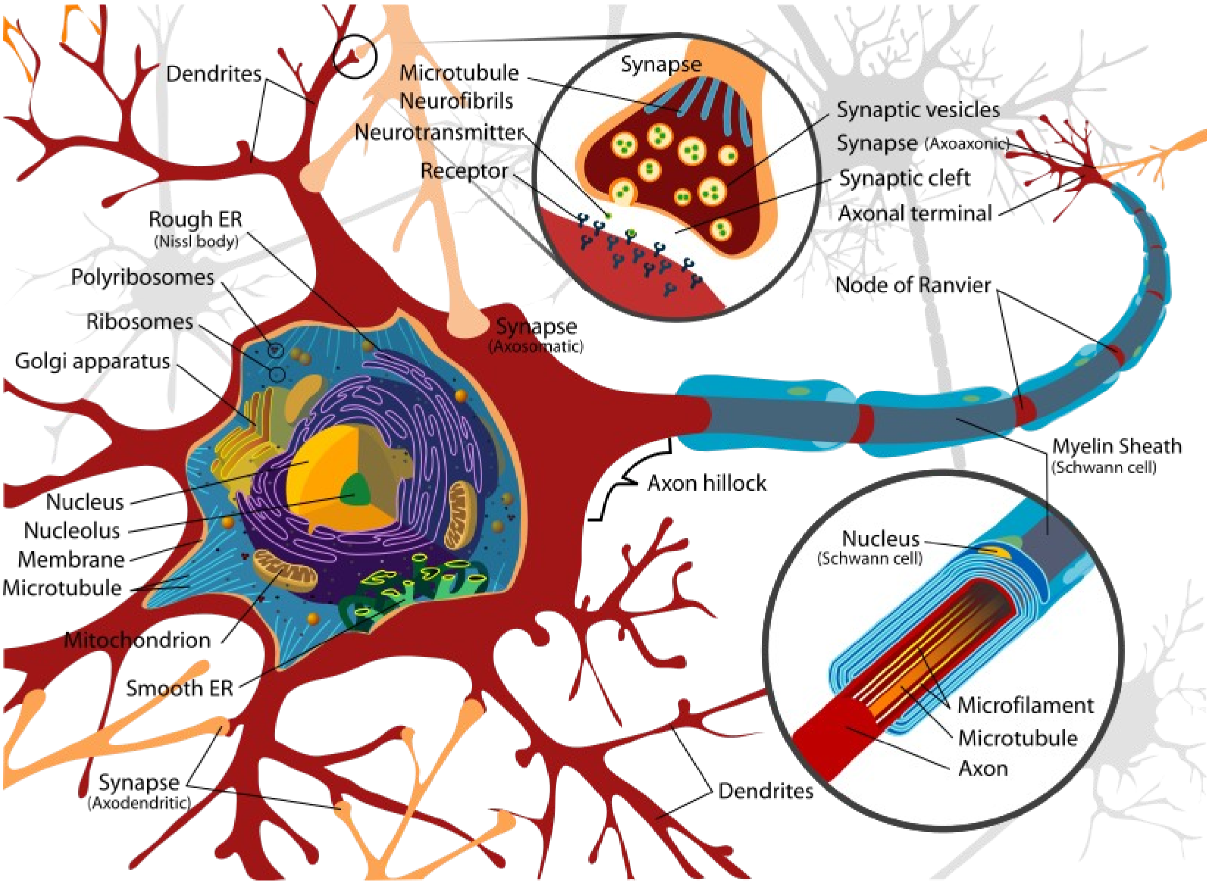
Here I discuss how the brain handles probably our most important task which is to remember events, places and times and make sense of it all. From this basis we can then use these memories to make informed choices about life events. Also, here I discuss the importance of Sleep and how that fits in with storing and organizing memories. Additionally, I will discuss how it has been discovered that some mental illnesses are related to poor sleep and memory acquisition.

Storing Our Memories In Neurons

Our Neurons store our memories which according to this theory are various inter-related message stories. Each Neuron has capacity. Studies have shown folks can store 7+-2 entries in their mind at the same time. Therefore we shall assume that a Neuron can contain approximately seven message entries in the Soma of the Neuron forming one message story.

The dendritic branches form the Message Algebra and the dendritic spines form the connections to other neurons forming the overall network. Note: The dendritic shapes indicate the type of Algebra that the Neuron type supports.

The axon is the cabling and the synaptic connectors at the end are the outputs while the dendritic spine form the message inputs.



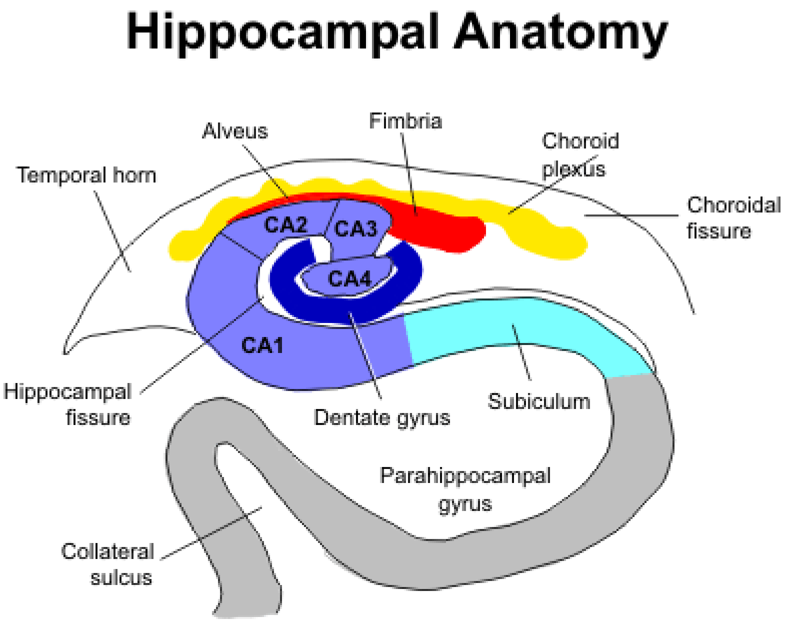
An Android should therefore use this approach for an actor cell which is the Neuron.

Storing Our Daily Memories In The Hippocampus

Our hippocampus stores and retrieves our memories. It is located near to our Thalamus and there are inter-connectors between the two systems. At the time of writing there is no definitive solution for how our memories work. In this theory there is a reasonably simple approach as to how it works which is loosely based on established research. The design I propose is simple and practical.

According to this theory, the Episodic memories which are the events which we experience each days are stored in our “Daily Storage” which is our hippocampus. The information or messages stories are stored chronologically in the hippocampus.

There are several sections which do this.



Each part is numbered and does a specific job. According to this theory and some established research the daily episodic events are stored here. However this area does not have unlimited storage. Mostly, it's for daily storage so the idea is to keep a days worth of information and at the end of each day we sleep and then certain genes / apps are switched on and the memories are uploaded to the prefrontal cortex where they are transferred, optimized, analyzed and the hippocampus is “cleared out” so when one wakes in the morning one feels refreshed.

An Android should have the equivalent of a hippocampus which stores daily message stories. It should also have a nightly process where its Message Stories are optimized and cleared out.

Sleep And Sleep Deprivation

According to this theory, the brain clears out the hippocampus each night of the daily message stories and starts again. These memories are analyzed and stored in the cortex where links are formed between the data.

One of the oldest forms of torture is sleep deprivation and it can have strange effects on people. Some folks have also tried it for charity and endurance testing. According to this theory the problem with not getting sleep when your body needs it is that the hippocampus will fill up and will then forcibly move these memories into your cortex.

In one such case a DJ who tried to stay awake for eleven days had periods where he entered a dream like state for ninety minutes and was in a semi-dream. According to this theory, the hippocampus is being cleared out to make room for new episodic memories while this happens.

DAILY MEMORY TRANSFER CORTEX OPTIMIZE

Normally this should be done when you are asleep. However if you force yourself to remain awake you will cat nap and people have been known to hallucinate as their consciousness is "switched out" during Memory transfer. The worst part about this idea is that the optimizations of the daily message stories will not happen and there is a possibility that the message stories will be stored in your cortex in a non optimized way and the Message Story linking may be very poor.

Therefore it has been reported that people who suffer sleep loss also suffer memory loss. Here this is due to the message story links being broken. This is not so much brain damage but poor message storing plus we must also realize these are the memories which we look up to solve problems so if the linking is poor then our decision making will be poor also. Therefore ones perceived personality could change depending on the types of message stories retrieved.

REM Sleep And Message Transfer

According to this theory deep sleep is when the body rests and rebuilds. When the body enters REM sleep the Hippocampus is emptied and the contents are uploaded to the cortex in blocks. This causes the eyes to move and the body enters a period of high sleep activity as Message Stories are moved from one location to another and optimized – according to this theory.

An Android should support this.

Mental Illness And Disturbed Sleep

It's been shown that schizophrenia is related to interrupted sleep. Why is it according to this theory? If one is trying to sleep and move ones daily memories into long term memory each night and it is faulty then it follows that the returned Message Stories will be faulty on lookup. Therefore one can be diagnosed as having “faulty thinking” aka mental illness.

Lookup And Store

The message algebra for MEMORY should have look up and store methods

MEMORY ALGEBRA

store chronological

backup chronological (aka REM sleep)

look up long term

store long term

optimize connections long term

switch out consciousness

switch back in consciousness

learn task procedural

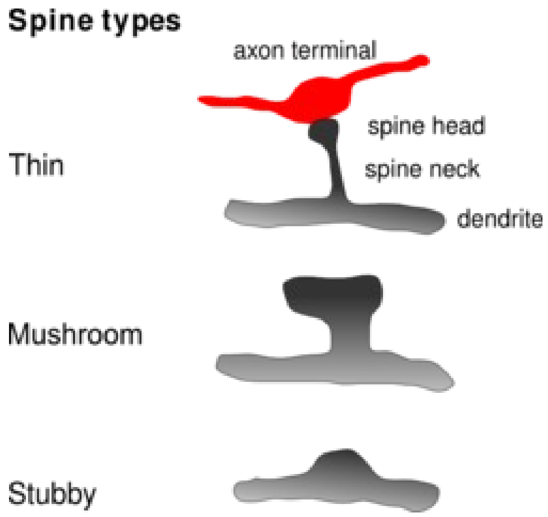
Procedural Memory

When we optimize our memories we gather together sequences of message stories under a type of task. For example someone might show us how to do a task in a location.

This is a “learn task” operation which is a sequential group of message stories.

Building Links Between Message Stories

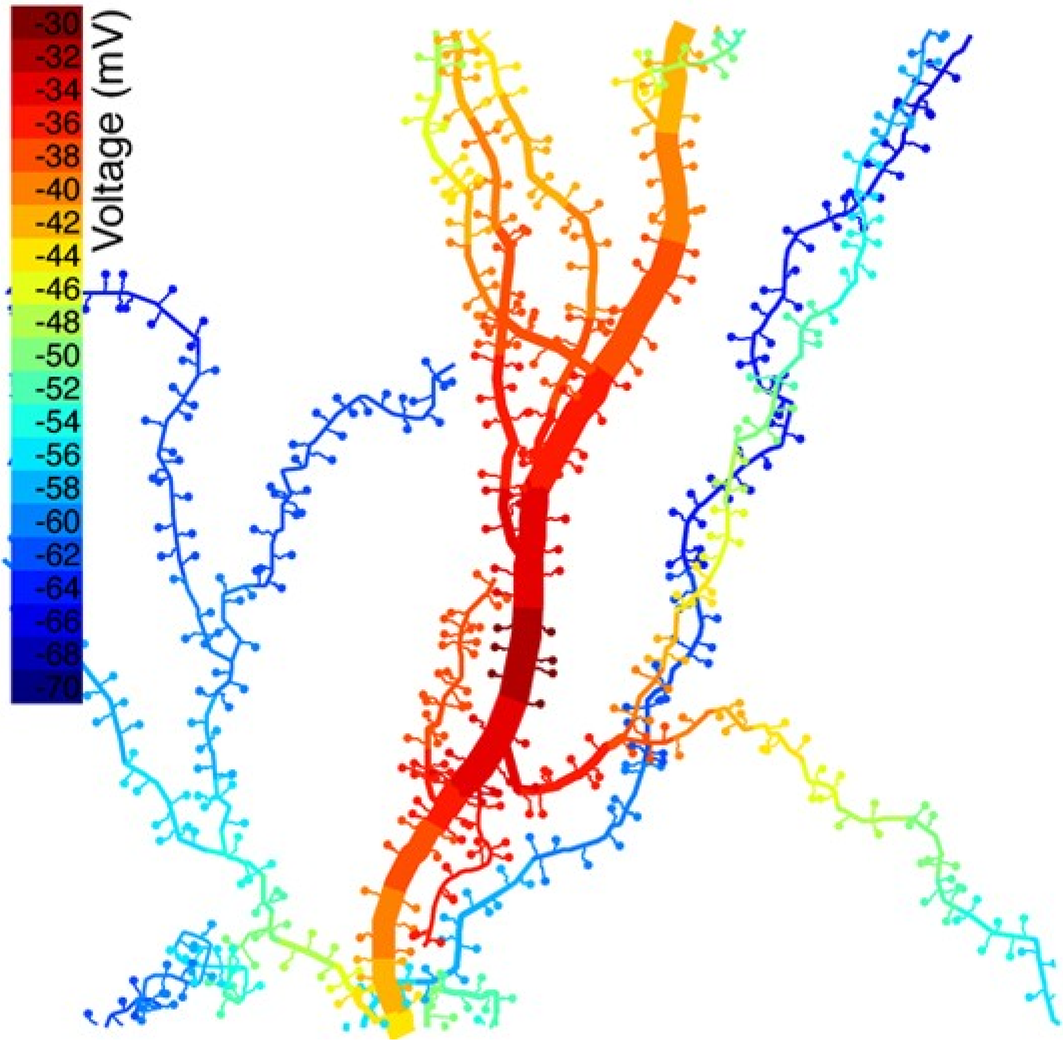
This is handled by the dendritic spines which number in the thousands per Neuron. These are "optimized" each night when one sleeps according to this theory. There are different shapes and sizes.



Voltage And Message Types

According to this theory the milli-volts that the Neuron receives from the dendrites maps to a message algebra message which the genes in the Cell understand. These voltages vary from person to person in terms of mapping which explains why some medicines work for one group of people and not another.

An Android needs to have some kind of dendrite mapping also.



Bibliographic Memory

Some people can recall exact dates for events in their memories. The way this works according to this theory is that the chronological backup of memory pairs the memory with the date

PAIR[

date

message story

]

This is a form of synesthesia which I have discussed before. Then when a memory is looked up one also gets the date.

This might be useful for an Android and the pairing of message types should be configuration driven.

The difference between this and other brains is that there is more storage required as there is more information returned. MRI scans have shown that the **Caudate Nucleus** is multiple times larger in people with this gift.

This area of the brain according to this theory is the Working Memory area which is associated with an SC room.

The Working Memory forms the context of the places and times in real time during the day.

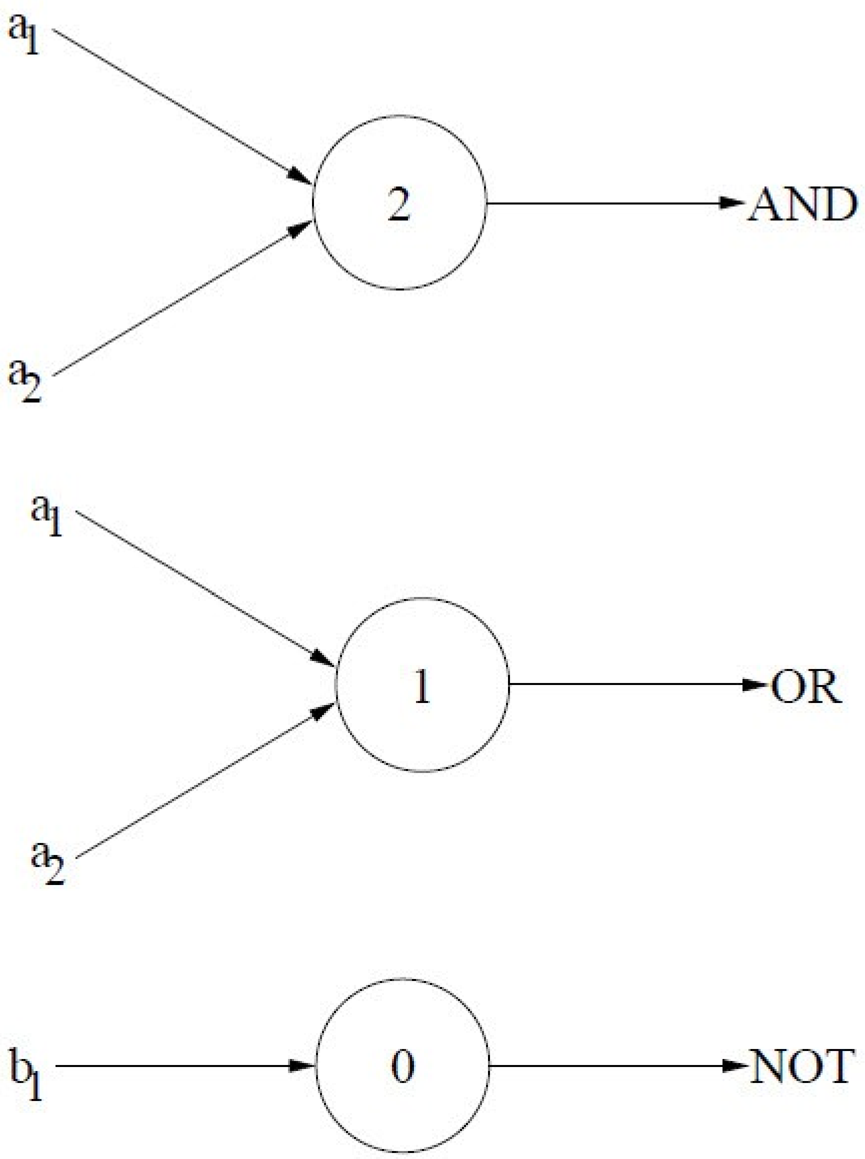
An Android therefore needs a large storage area to support a feature like this this type of recall.

Chapter 12 Logic, Reason And Math

Here I discuss how the brain handles logic, reason and math. These are advanced feature of the Messaging System and I will describe how our Neural Net achieves this according to this Theory.

Implementing AND, OR and NOT logic

In 1943 McCulloch and Pitts realized that the brain must have some kind of mechanism for AND, OR and NOT operators because they form the basis of logic. They examined neurons and came up with a model.



This approach has been refined over time by Minsky who created the Perceptron. In the 1980's an algorithm called neural back propagation determines the weight of the connections and determines the correct path through a neural network. Modern pattern recognition uses algorithms which learn and then use the weighted network to make comparisons. The idea of AND, OR and NOT is regarded as being deduced by the Neuron Weightings which are trained to “learn” and became less important.

In this theory, the design for an Intelligent Machine requires the use of Neural networks for pattern recognition in part (but producing specialized Message Stories) but the **logic and math aspect of the intelligence and logic is based on an Advanced Messaging Algebra using Traditional Set Theory**. This is how this theory implements AND, OR and NOT and other logic operations which are deduced by MESSAGE STORY set theory.

Let's take a practical example.

In the memory storage area we store MESSAGE ALGEBRA STORIES.

A bird has wings and can fly and has a beak.
A Penguin is a bird but cannot fly

Question: What is the difference between a bird and a penguin?

Each sentence is a message story in memory.

MESSAGE STORY
BIRD
CAN FLY
HAS A BEAK
END MESSAGE STORY

MESSAGE STORY
PENGUIN
BIRD
CANNOT FLY
END MESSAGE STORY

MESSAGE STORY REQUEST, DIFFERENCE BETWEEN

BIRD

PENGUIN

END MESSAGE STORY REQUEST

The message stories are retrieved and we use MESSAGE ALGEBRA SET THEORY to find an answer.

Here we can do

UNION

INTERSECTION

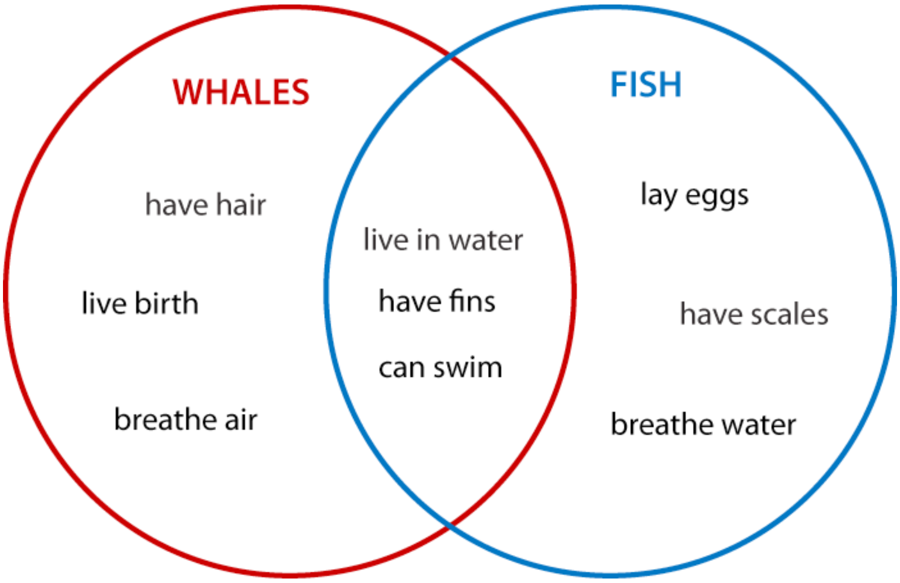
COUNT

IS AN ELEMENT OF

SUBTRACT

and so on.

These are your typical Venn diagrams one learns in school. Each of these bubbles can be considered to be a Message Story stored in the brain. There are two here for example.



In our case, we look up the message story for BIRD and for PENGUIN.

If we subtract BIRD from PENGUIN (which is the difference) the MESSAGE STORY we get is CANNOT FLY. Therefore the difference is : **A Penguin cannot fly**. Each Venn Diagram represents some kind of MESSAGE STORY and from these operations we can start to build LOGIC DIAGRAMS and OPERATIONS.

Therefore you don't need Neural networks to solve logic problems according to this theory, you need to store the MESSAGE ALGEBRA as MESSAGE STORIES and then perform SET THEORY OPERATIONS on them. This can be for any kind of MESSAGE ALGEBRA like Sound Recognition and Vision Recognition so long as the information is encoded as Message stories.

According to this theory, the HOS has a Core Algebra for Message Stories based on Set Theory independent of the location we are born into so it gets mapped to the Language Locale.

An Android should have this as well.

Working Memory And The Caudate Nucleus

A pretty important part of the whole human memory mechanism is Working Memory. This can also be called Short Term memory for information that we store briefly such as phone numbers and the like. Research shows the area in the brain where this is could be the Caudate Nucleus as an educated guess. However no matter where it is located we still need this storage according to the theory.

Working Memory forms the core of who we are.

In this theory, each SC Room has a Working Memory storage area.

Our RR has one and so does our Dream Room and any other room such as ones where we predict and design.

Why do we need it?

The reason we need Working Memory is to form context.

Consider hearing a word that sounds like either 'hare' or 'hair'. How do we know which is the correct one. The working memory contains Message Stories like.

[

FIELD

WILDLIFE

]

If we look up field wild life we find links to 'HARE'. Therefore working memory determines the context of our situation based on the last moments.

Memory is best chunked. This means we are good at remembering message stories which contain between 7+/-2 entries. So this works out at between 5 and 9. Therefore phone numbers tend to be in around seven digits long (but it does not have to be numbers).

[1,2,4,5,7,8,9]

Harder to remember are longer ones such as

[1,2,3,4,5,6,7,8,76,5,4543,3,3,4,45,56,6,67,7,87,8,8]

Therefore we tend to break them out into blocks of say three numbers and create a sequence of message stories. The brain is better at remembering these.

chunking {1,2,3,5,6,7,8,9} = easier {1,2,3} union {4,5,6} = {some phone number {1,2,3} union {1,2,3}}

Remembering By Repeating

We repeat unfamiliar numbers to ourselves to keep it fresh in our mind. Therefore message stories with no immediate connection to other message stories in working memory are dropped after about 10 or so seconds. So we repeat them to ourselves and feed the numbers back into working memory.

Remembering By Deep Learning

When we study we create message stories and these are looked up in long term memory to see if there are any associations.

Story A [

Medical Procedure A

Illness B

Hospital C

]

Lookup hospital C.

Returns

Story B [

Hospital C

Doctor D

Specialization Illness B

]

Therefore a connection is made in term of a Union of Story A and Story B.

Union Story A, Story B

Store Long Term Memory Story A

Store Long Term Memory Story A, Story B

Therefore we learn Story A and it has a link to Story B and its placed in Long Term Memory.

Studying requires that we build links to our existing stories and expand them but linking in other stories. Schools typically work on this principle by building on the basics (aka Basic Stories)

If a student crams before an exam which is a popular technique basically one fills up Working Memory with lots of Message Stories related to the topic in question.

Also, the results of logic requests are stored in the Working Memory.

We might walk into a room and ask

What is this place?

We will send a message story to long term memory to look up what we are seeing and find a match e.g. it's a hospital.

So

SC.RR.WORKING MEMORY is pretty crucial.

If we are asleep, according to this theory our long term memory organizes our Message Stories by locking it down and we work on Working Memory only.

If we are in our Dream Room and our long term memories are unlocked, we can store some of the Dream Working Memory Stories in our Long Term memory but our SC RR Working Memory may find that these message stories make no sense to us.

It's a key component for an Android to have Working Memory as a place to store context in a Real World environment and have somewhere to store logic results.

Note also that Motor Message stories for say riding a bike are stored in the Motor Message Storage part of the brain. It contains things like the pressure we need to place of a pedal which is automatic and so on but the steps to ride a bike are stored in Long Term Memory so we can teach someone how to ride a bike verbally.

Remember that Message Stories can pretty much contain any kind of Message Type and Algebra.

Building Strong Links Between Message Entries

According to this theory, the Neurons offer an interface to their message entries via the dendrites. If one thinks of two neurons connecting then one can imagine two Venn diagrams which have intersections. The intersections of data is where the connections are the **strongest.** Conversely, where there are no intersections of data, the message connections are less strong. According to this theory, the Denrite s offer an interface to each message entry but it has an upper limit on how many it can store.

Layering the Neurons

Each layer of Neurons can therefore be thought of as a series of interconnected Venn diagram layers where each layer moving in one direction offer more complex pattern recognizers based on the data in the Neurons.

Techniques for Improving Memory by pairing with Visual Messaging

An old but successfully mechanism used by memory experts is to pair message stories such as groups of numbers or lists of some kind with Visual Messaging

```
PAIR[
MESSAGE STORY
VISUAL MESSAGE STORY
]
```

Why does this help?

The answer according to this theory is that the brain has a SC Room with working memory. The working memory does not last long but if you pair this information with Visual Messaging then it will be stored in the Visual Room and persists longer. Therefore folks who can remember large lists are essentially moving around a Visual Room and have labeled these visual keys with non visual information. The brain is highly evolved to handle visual information and it's a key part of its survival information. If one loses a phone number one can find it again. If one is chased by an animal and one loses its location one can end up being consumed by it so the brain values Visual Information.

Story Telling And The Ability to Create Memory Stories Using Imagination

Before humans could read or write there were people who passed on information in the form of story telling. Even in the world today there are people who learn texts in great detail by repeating them. In ancient times, songs and stories would have been told to the next generation and this required people with very good memories.

Remembering lists by stories is an extension of this where someone with this innate ability can create Memory Stories of great length and recite them internally word perfect.

Therefore imagination can be seen as the ability to construct message stories based on incoming information.

```
CREATE MESSAGE SEQUENCE
SEQUENCE [
PAIR A,B
PAIR B,C
]
SEQUENCE[
PAIR D,E
PAIR E,F
]
STORE MESSAGE SEQUENCE
```

Pairing Visual and Non-Visual Messaging To Memorize Lists

Visual Messaging is stored longer by the Neurons according to this theory. These messages build images in our minds which are rooms associated with our SC.

Normally, we associate non visual messages with visual messages e.g. the message for a bridge and the words for a bridge.

```
[
VISUAL ALGEBRA BRIDGE
ENGLISH WORD BRIDGE
]
```

So when we see a bridge the word Bridge is associated. Therefore when we look at a scene all the images have associated words. The brain is really good at this and needs to be fast as it's part of our survival instinct.

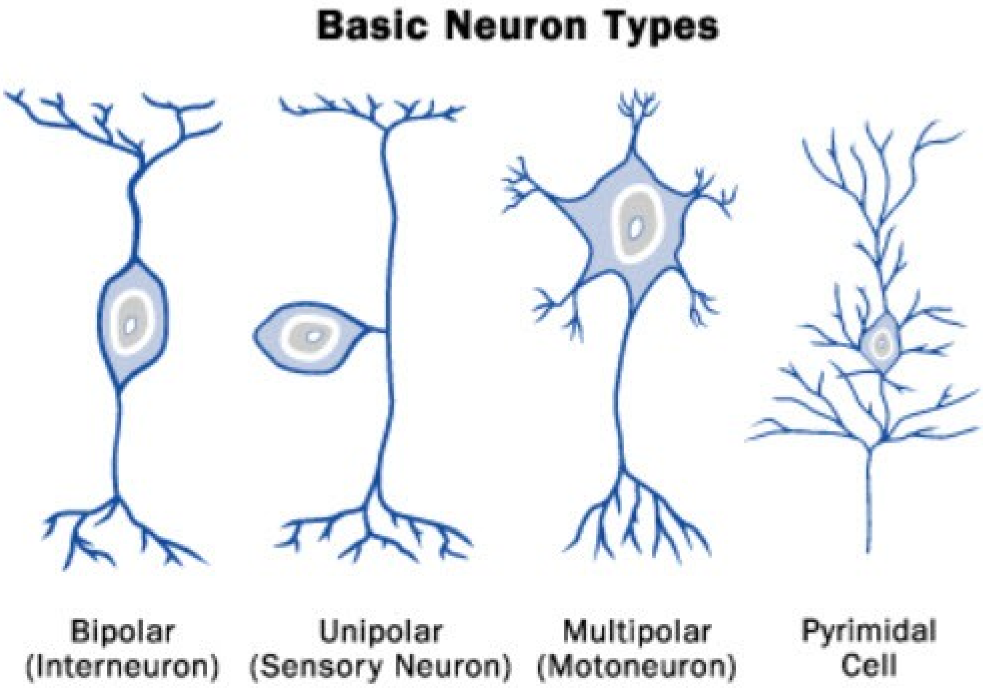
If however we just get a Word which does not have a visual pairing, the Neurons in our working memory do not value this highly and it can be dropped pretty quickly.

Memory Stories can be created with Visual Messaging so that difficult to remember lists are paired and are easier to look up.

This is basically the same as what we are doing normally from day to day so we take advantage of this ability of the brain.

Visual Messaging lasts longer in the SC and Working Memory therefore and can be paired.

Mapping Neuron Types To Message Stories



If we assume that a dendrite maps to one entry in a message story and the Soma contains the message story we can decode neurons as follows. Ones that are not covered here can be interpreted in the same way.

Bipolar (Interneuron) has two messages in its message story. This could be a store-and-forward design pattern.

Unipolar (Sensory Neuron). Here we have one message which is shared by both dendritic tree. This is how we can copy one message into two messages which are the same.

Multipolar (Motoneuron). Here we have a Message story with many entries. In the diagram there are five entries which can be operated on and sent out.

A pyramidal cell is shaped like a pyramid where there are inputs which are joined together. According to this idea these types of Neurons which are present in the cortex join pairs of Message Types like say Video And Audio messaging into one Message story.

e.g.

(Pyramid Neuron)

PAIR[

AUDIO MESSAGE

VIDEO MESSAGE

]

The output is a message story type which is a combination of Message Types or a Type Aggregrator. If the theory is correct we should find these types of Neurons in the Thalamus and it appears based on some research that they are present there.

An Android should implement these design patterns for its Neuron implementations.

Processing Logic And Reason In Working Memory

In the theory, logic and reason is processed in Working Memory. Long term memory storage and Daily storage in the hippocampus mainly associated data together. The act of asking a logic question and solving it happens in the Working Memory area where Message stories are retrieved from the daily memory and long term memory.

Ask Question

Covert to Set Theory equivalent

Look up message stories in the Daily Memory

Look up message stories in the Long Term Memory

Apply set theory operators to the returned message stories

Build the result

Send to the language centers

This is how an Android should do it according to this theory.

The Basics Of Math And the Brain

In order for the brain to understand Math one must first understand the idea of counting and sets or groups of items. This requires that the brain supports specialized operators according to this theory.

First the brain must support the idea of counting a set of objects.

```
COUNT[
MESSAGE STORY[
DOG
CAT]]
```

This should return two items for example. Typically studies have shown that when people are shown diagrams of objects like circles and told to total them up, people with normal math abilities gather then in group of several items and add them that way. People who have little or no math abilities tend to count one at a time and do not have a good idea of what counting means. If one has natural math ability according to this theory, one of the abilities is to count the set of objects. This is a basic function of their natural set theory COUNT operator. Therefore when they see groups of objects their neurons can count how many items are in the set. Normally this is limited to a few items which is the upper limit of a neuron according to this theory.

Additionally, when people lose the ability to do math or have none, studies have shown they also do not have good abilities to visualize.

For example: Point to the person's left shoulder.

To do this one must mentally position ones self as the person you are facing and imagine where their left shoulder is.

To do this requires VISUAL ALGEBRA operations

MOVE

TRANSLATE

ROTATE

Many math concepts deal with operators like this for charts and the like. If ones brain does not support this, then one can lose math abilities as charts and properties of those charts will make little or no sense. This is centered in brain regions to do with this.

An Android should support this.

Messaging Algebra And Mathematical Algebra Systems

The messaging algebra needs to have an associated app. It's important that an Android or even a human can solve problems using rules. Therefore we need to use Algebra to solve problems and not just blindly calculate.

At the time of writing there are a few Math Algebra systems.

Mathematica

Maxima and so on.

At the time of writing Maxima is open source and could be integrated as it is open source.

MATH MESSAGE[

```
SOLVE  3*x -2
]
```

Ideally we need rules based messaging which maps the operation to the sequence of steps forming some kind of logic rules similar to a computer program.

Chapter 13 Learning And Mapping Information

In this chapter I will discuss the design for learning information and mapping our neural networks to a locale which has a language of a particular type and written symbols. If one wishes to localize an Android to a particular location then this needs to be setup. I will discuss how the brain does it according to this theory and then map that to the Android design.

Underlying Abilities At Birth

Our DNA has a series of message types which are independent of the language and symbols of the locale we are born into. However we need to learn the rules of the place we are born into. The first set of rules is language which is generated at the societal level. Later we learn reading and writing. According to this theory, the words and language we choose reflect the underlying Message Algebra at the core of our HOS.

Society Based Language Formation

Each group of humans or even some species develop their own language. Research on birds taken from the main group of birds and isolated find that they develop their own songs and each successive generation refines the song until it matches the one already in existence. This provides a clue that the bird's brain is somehow aware of the pattern of the song and it is not just random chance.

Also students are shown images with random non English words. They are asked to write them down again mostly getting them wrong but these words are then used with the next set of students. Over time, the random language evolves to have some rules about how the images relate to the words.

According to this theory the reason for this is that we have an underlying HOS Message Structure and the words we come up with are crude attempts to map these words to that Message protocol.

From this we get the grammar and syntax of a language which is a reflection of the HOS Message Algebra.

Therefore language acquisition is essentially a pairing of sounds with HOS message algebra.

At its simplest we pair images with sounds which form words

```
PAIR[
OBJECT
SOUND
]
```

Over time we pair multiple words and form sentences. From here we generate logic and grammar.

```
PAIR[
 WORDS
SOUNDS
]
```

Speaking

Children grow their vocabulary from one word to two and then onto sentences. By the time they are five they can know up to ten thousand words. According to this theory the new learned sounds are networked from the HOS Algebra as the brain develops creating new layered patterns. These layered patterns represent a child learning the locale usually via some kind of parent or care giver.

The Evolution of Speech

Research has shown that our vocal chords are nothing special. A dog for example has the equivalent of a larynx. However it just barks. So how can we define this special gift that we have according to this theory? Let's look at the type of messaging a dog and a cat make when they make sounds.

Pet Experts for dog.

Continuous rapid barking at a mid-range pitch: “Call the pack!”

One or two sharp short barks at a mid-range pitch: “Hello there!”

Single sharp short bark at a lower mid-range pitch: “Stop that!”

and so on.

Pet expert for cat

Short meow or mew: Standard greeting. “Hello!”

Multiple meows or mews: Excited greeting. “Great to see you!”

High-pitch RRRROWW!: Anger or pain. “That’s my TAIL you just stepped on!”

What this shows in the main is that animals mainly have sounds associated with emotional algebra.

Animals don't make sounds for objects. For example if you show a dog a ball that it will chase it does not make a special sound for a ball.

ANIMAL PAIR[

EMOTION

SOUND]

On the other hand human beings have developed a wide range of sounds for objects and for logic and reason. Young children start by pairing words with sounds.

PAIR[

WORD

SOUND

]

Later we have simple sentences.

PAIR[

SOUNDS

SENTENCE

]

Therefore one can imagine early humans which were like us developing a form of synesthesia between sounds and our underlying Message Algebra in the SC located in the Thalamus. This apparently happened about fifty thousand years ago. Before this we were probably like dogs perhaps where we only made noise for certain Emotional Algebra e.g. happiness, sadness, anger etc; To a certain extent one can think of an angry person using an old form of our sound mappings while someone reading a book is using our newer mappings. According to this theory the synesthesia was a mutation in our population which gave rise to us in our present form.

PAIR[

MESSAGE ALGEBRA

SOUND

]

It may have been as simple as that in the SC and one which made all the difference.

Even today we see new synesthetic mutations connecting days with memories and visual attributes with letters for example.

An Android should have this Message pairing with sounds as it is key to speech acquisition.

Speech Circuits

The circuit is as follows

General Messages are paired with the Speech and Language Centers in the SC from which Grammar Based Sentences can be formed. (This requires a locale based mapping.)

They are passed to the Language Center

This is then passed to the Speech Center

The Auditory Center also takes in sounds and they are then stored in the Daily Memories

The Vision Center takes in images and they are stored in the Daily Memories

Language And Grammar Based on Underlying Message Grammar

According to this theory, we assume that each Neuron stores several entries. In some cases we pair different types of signals such as Audio and Visual messaging.

Our working memory has goals and these may involve having to interact with others. To do this we need to construct some kind of verbal sentences in the speech centers to convey an idea.

We may need to look up some kind of Visual Memory in our Working Memory.

Visual memories therefore must contain **time information** for the present, past and future.

They must also contain some kind of person or subject, possibly a visual action and so on.

Therefore no matter what language we have they must all cover the basics of the Message Algebra in our brain for Visual situations.

Let's take the example of Mandarin Chinese and English.

The grammar rules for a complex Chinese sentence are

Subject + When + Where + How + Action

This maps to a Chinese Sentence Neuron

Chinese Grammar Message Order[
Subject,When,Where,How,Action
]

e.g.

English : I study English at School with John tomorrow.
Chinese English Order: I tomorrow at School study English with John.

The Subject is itself another Neuron and so on.

Obviously the symbols are different but the idea is that the Visual Grammar in the brain contains.

GENERIC VISUAL MESSAGE STORY[

Subject

When

Where

How

Action

...

]

Everyone is born with this. It's up the culture in question to map this information to the locale.

MAP CHINESE LOCALE GRAMMAR[

GENERIC VISUAL MESSAGE STORY

Chinese Grammar Message Order

]

Therefore if we have an Android and we want it to understand Mandarin Chinese we need it to have the mappings from the generic messaging algebra to the locale algebra for cases like grammar when it comes to sentence construction.

To do this would require specialists to build the mapping for each language and locale from the core messaging. The idea is to move from the idea of language to language translators to Core Message mapping specialists. They'd still be needed of course but for an Android this would be the approach and would be cross disciplinary.

Example Question And Answer

Let's work through a simple example to show how this works in a simple way.

Question: What did you do yesterday?

→ Convert question to message algebra in SC which stores it in working memory

Lookup daily messaging for yesterday

returns result messages to Working Memory

Visual Algebra Dog Walking

Emotional Algebra Dog Finds Injured Bird

Auditory Algebra Injured Bird Noise

This is then mapped to English/locale language

Passed to Sentence Center and then to Speech Center

I brought a dog for a walk

**The speaker makes a sad face due to emotional algebra

My dog found an injured bird

The idea here is that the English sentences are not stored in the Long Term memory but rather message types which are stored/retrieved and then converted to the local language.

The underlying language of the HOS is called HOS Language or **HOSL**. In the case of an Android we'll call this the OS Language or **OSL**.

Memories of times past, saving message memories in DNA Firmware

People have goals, for example

Reproduce

Find mate

Build home

Prosper

According to this theory, many goals have saved message stories from times past passed down through each generation. These are encoded message stories in the DNA . Using a computer analogy these are the firmware stories of the body. These are important stories which for example describe to a baby spider how it must climb to the stop of an object and spin a thread and let the wind take it. This is more typically called “Instinct”.

Other examples are

Bats return to cave for breeding

Birds migrate seasonally

Laying eggs on same beach

Spawning in same river

Some birds have geomagnetic senors for aerial routing

sensor_geo_magnetic long,lat

These paths are revisited, refined and restored if required.

These repeat paths are stored in memory

An Android should have some firmware stories which means that certain information does not have to be taught such as utility behavior.

Messaging and pathways both Internal and External

All message types have an associated pathway which can be derived internally in the mind or externally using some mechanism.

Therefore message pathways can be internal in the brain or external based on a scent trail for example and serve to fulfill some kind of goal like finding daily food.

Examples:

Internal pathways e.g. human brain (connected neurons, messages, SC, visual room)

External pathway e.g. ant (scent trail, message)

Each neuron is like an "immovable ant" which creates a pathway to and from other neurons, similar to when ants cluster together to form a ball of connections.

Therefore human roads are our equivalent of scent trails

Street signs are messages

The roads are the pathway

Our goal might be Buy Shopping or Goto Work

We can build a virtual copy of this in our minds

Ants typically just gets lost when the scent trail is removed but can rebuild

It appears like there is no internal model in base worker ants

An Android should have both internal and external messaging models.

Messaging distribution models

Different life forms distribute the messaging path in different ways. An Ant typically externalizes it whereas a human being has more internal processing. Let's discuss the different cases.

Basically what we want to do is:

Find a goal path and improve on it

Ants use the strongest scent from multiple ants to form the preferred path

This is like back propagation of neurons

So to satisfy a Goal there is

1. A goal path

2. Message for each part of the meaning of the path[s]

This is how the brain works according to this theory

Spiders on the other hand have evolved an internal message path for GOAL FEED which can attack the internal messaging of its prey; therefore it can paralyze them with venom.

Ants distribute the messaging across the colonies' rigid structure and it is mainly externalized for GOAL Feed. Aphids and other bugs are herded. Ants are socialized and controlled by external scent messaging and their internal messaging paths are limited unlike humans. Therefore the Ant Colony behaves collectively like what we would think of as a “Body Function” where this decision making is internalized in a person. Here, for example, they behave (using Messaging) like a liquid.

Spiders extend their body structure via webs and attack with Venom - the web is a disposable extension of their body. The Spider disrupts the internal messaging of its prey.

Bees communicate with each other by doing a Bee Waggle Dance. When a Bee finds a location for honey, it communicates the location to the rest of the Hive by means of a directional dance and a set of buzzing motions which the other Bees understand. This is another type of inter-Bee messaging protocol. Notably colonies of Bees have been disappearing and logic tells one that if certain changes to the environment affect the Bees messaging system particularly inside the Hive itself producing a chemical message or signal that matches LEAVE THE COLONY then the Bees will leave it even if there is not an actual problem.   (I realize this is a complex issue but it would be good to understand the Bees Messaging Protocols is the general point).

Mammals use their groups to distribute certain survival messaging like running suddenly to alert the others to run also.

Humans optimize individually but have specialized group messaging e.g. different tasks but do not rely on a single queen ant but delegate that role to a leader who handles that type of leadership messaging . Each country handles leadership delegation in different ways but the point is that all countries have some kind of leadership structure to which the general population looks to for leadership guidance.

Associatively humans tend to form groups of twelve or less where they typically ask : who's the leader? Then groups report into groups and so on in a hierarchy. The higher in the hierarchy messages are passed down and therefore the messages impact more individuals and are therefore are deemed more “powerful”. A squad of people is around ten and most likely dates back to our hunter gatherer times.

An Android should probably have some kind of assigned leadership status or be aware of who to report into to take orders and understand human group dynamics. It should probably be assigned some kind of group or unit designation even if that is to a family or shop etc;

Unfamiliar Life-forms

Side thoughts:

What we have not seen is a life form which samples DNA and copies / integrates it to copy predators. A cross between spider, human, bee and ant and plants and attracts prey with sexual / food signals could possibly exist on other more evolved worlds. Here we enter the realm of SciFi.

It could be that origination of life is based on a super messaging life form we have not seen on earth or that the messaging has grown from life originating at the nano level "bubbling up" so to speak.

Androids would be useful for initially scouting out planets which are unfamiliar to us.

General pattern to intelligence in this theory And Definition

What is life? : That which contains internal and or external messaging paths with different distribution models and strategies for evolving actionable message types and physical forms to achieve necessary goals.

Therefore all life

Finds a message path to GOAL e.g. food, mating, home making etc;

Record it as some kind of message story (scent based, genetic, verbal etc;)

Make sure it is readable by others of the species

Also support some internal messaging stories which also satisfies one or more GOALs

Then follow the path and refine using improvement messaging - re-communicate etc;

An Android should have clearly defines Goals and some firmware Message Stories when it is booted up; e.g. know how to power itself up and how to be sociable for example and report on its status.

Local Customs And Etiquette, Social Interaction Protocols

In human society there are different rules for Meeting, Greeting, Eating, Talking, Mating, Doing Business, Social Cues, Body Language. For the most part an Android should be aware of these in its Firmware Stores and Locale programming.

This can be covered under Social Rules messaging.

For example, how to address a person and exchange details.

Territorial Signaling

In the animal kingdom Birds make sounds and animals growl to mark territory.

An Android should not be aggressive but should have some way of warning others in cases where the assigned group has some issues.

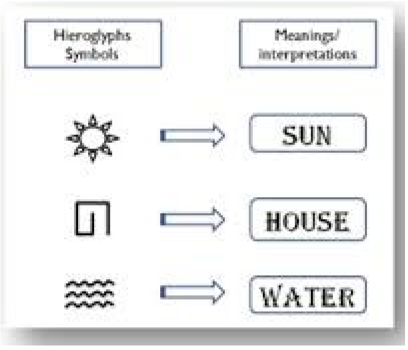
Reading and Writing

According to this theory written language is our best attempt to reflect our internal Messaging which originates as the spoken word. The mapping to symbols requires that we use our creative abilities and therefore we find that there are many different approaches to language but all are trying to represent the spoken language which builds on our internal messaging.

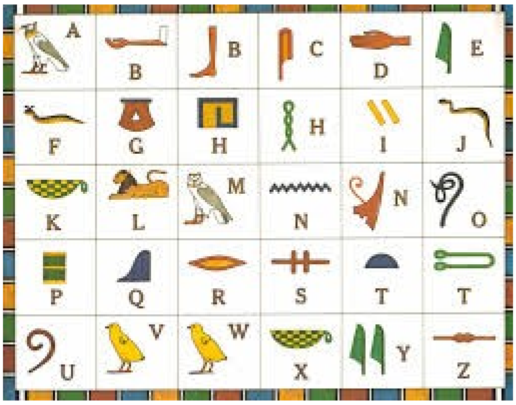
Here we have some of the better known approaches to writing.

Pictograms were used in Mesopotamia and written on clay tablets.

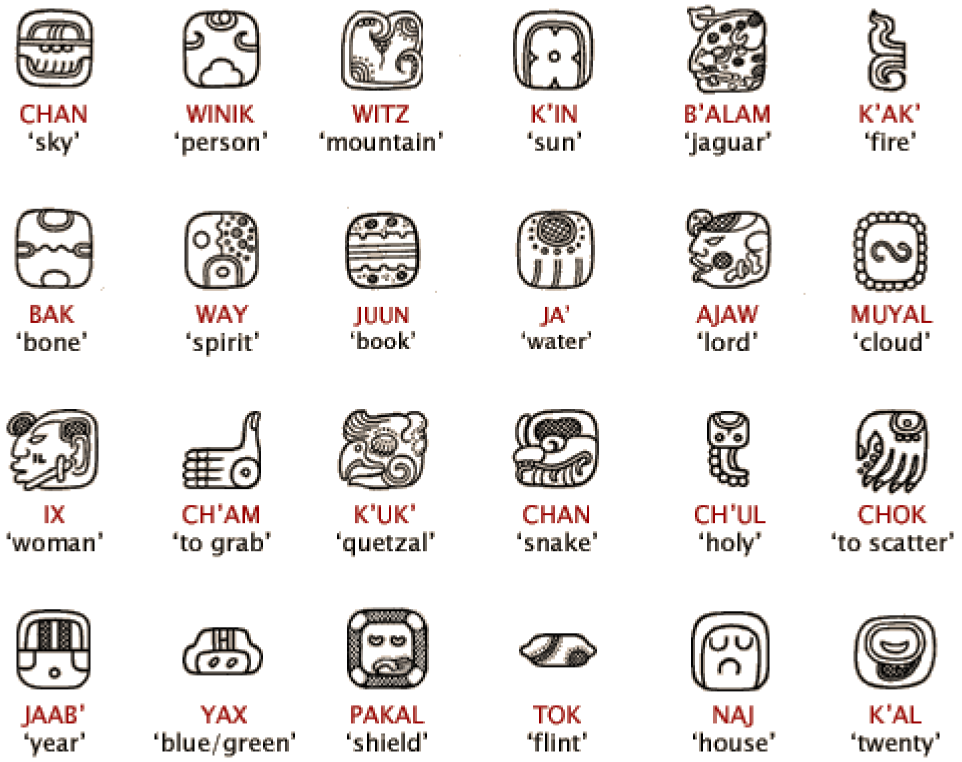
The were used to map simple concepts.



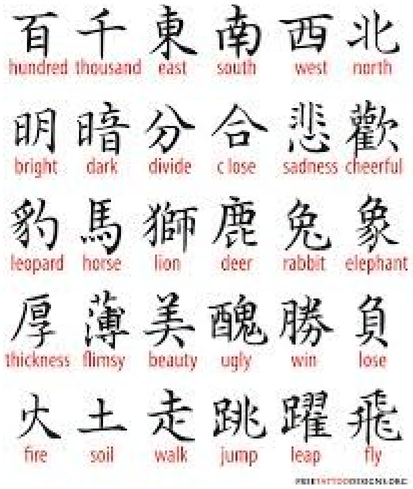
The Egyptians had a phonetic piece to their symbols.



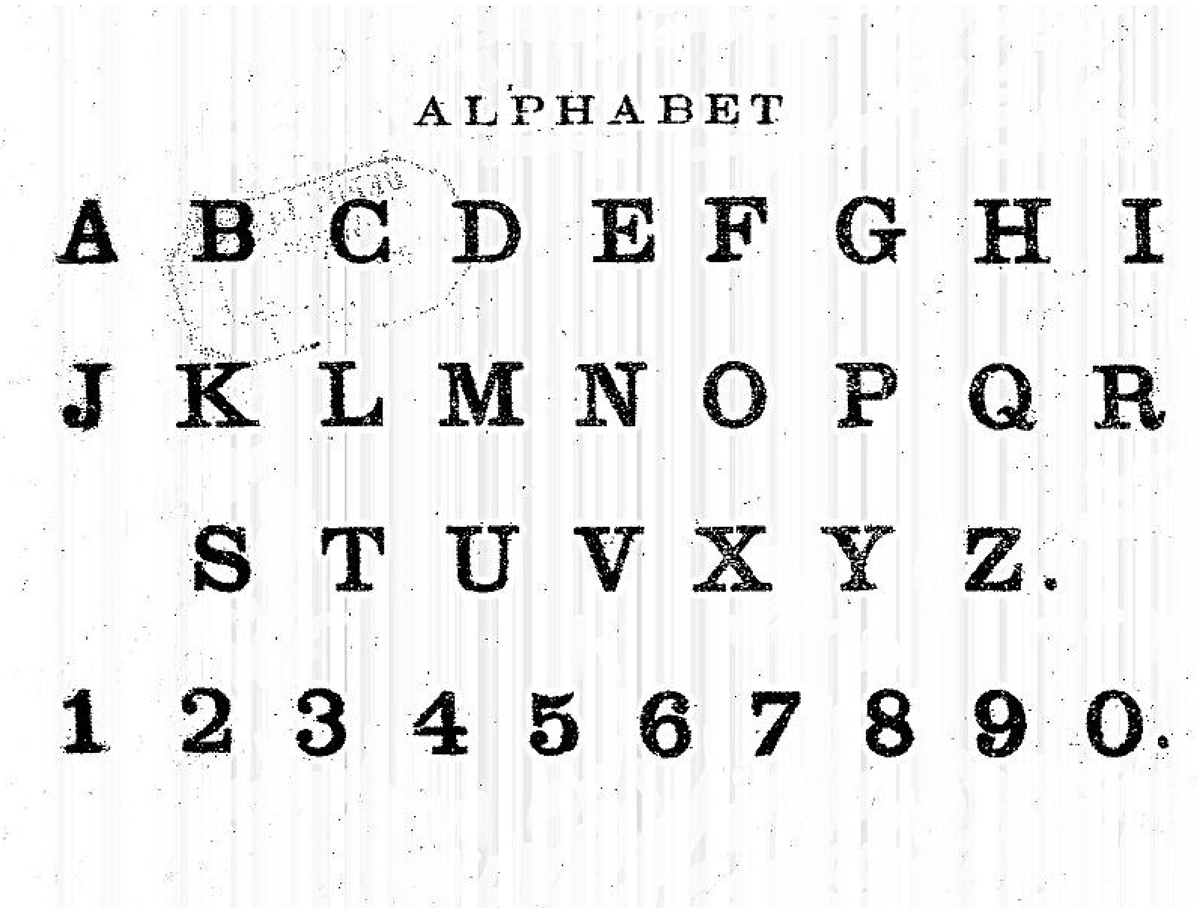
The Mayans used Glyphs which related to words in some cases.



The Chinese use symbols for words, concepts and sounds and the list of symbols is very large.



In Western Europe and surrounding areas we have a Phonetic Alphabet from which we form words and sentences.



ALPHABET

A B C D E F G H I

J K L M N O P Q R

S T U V X Y Z.

1 2 3 4 5 6 7 8 9 0.

Typically speech comes first and then one needs to account for thing using numbers and record them. One also needs to record the spoken word and so one approach which is popular is to create symbols that match sounds. Also as the written word develops beyond speech we need to extend the written language to handle all the ideas of the internal messaging in one's mind. Therefore all language and speech evolves to describe the underlying messaging and the changing environment.

MAP[
WRITTEN WORD,
SPOKEN WORD]

MAP[
SPOKEN WORD,
INTERNAL MESSAGING]

Over time we also develop Visual Symbols in the language to match our underlying Visual Algebra.

For example, we do not need words for these as they are part of our Emotional Algebra which is paired with our facial expressions for example.

MAP[

VISUAL IMAGE,

INTERNAL MESSAGING]

An Android should support these mappings from the words to symbols and also support visual symbols which match underlying emotional algebra and so on.

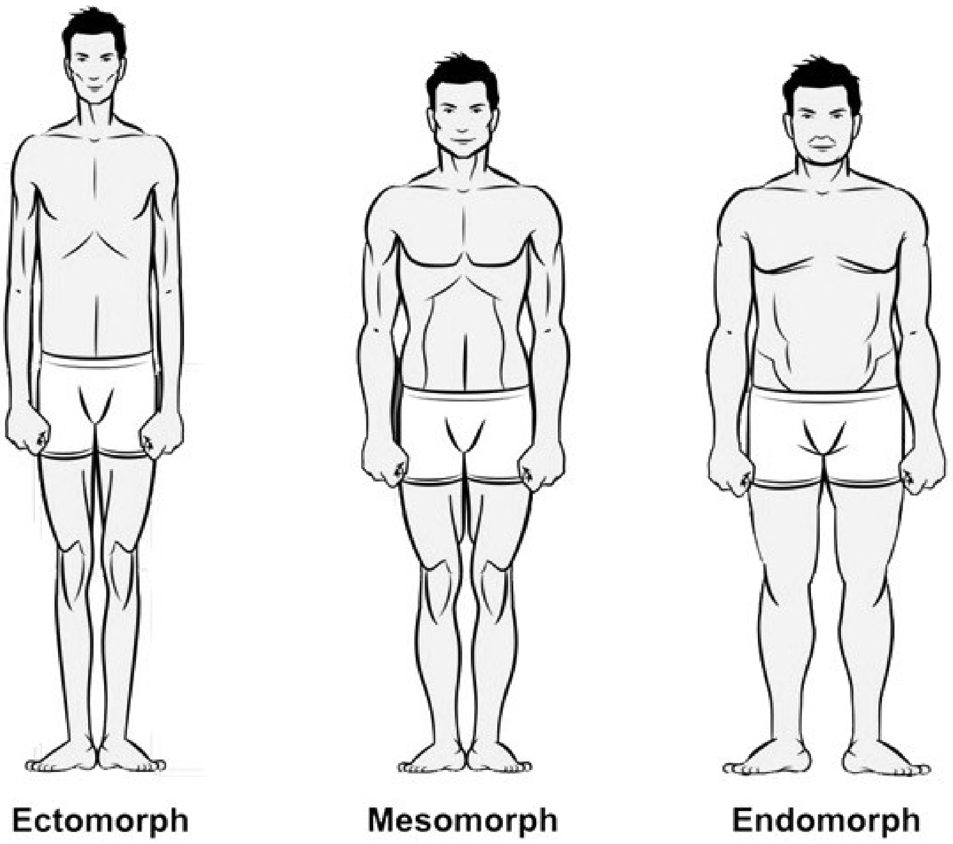
At present on Earth the closest we have to a universal mapping to all languages is Unicode so one can imagine the INTERNAL MESSAGING in the brain implementing some kind of OSL Unicode.

Chapter 14 Body Structure And Muscle Type

Here I discuss body structure and the types of connections.

Body Types

There are three accepted body types.



An Android can have any of these depending on the nature of its duties.

Muscle Fibers

Type I, better known as slow-twitch fibers, are the body’s primary method for less explosive, sustained movements. They do not contract forcefully and thus require less energy to fire, which makes them well suited to long distance running.

Type IIx are best known as fast-twitch muscle fibers. These are the muscle fibers primarily responsible for fast, explosive movements like sprinting. However, they lack the endurance-boosting ability of slow-twitch fibers and can only be used for short periods of time.

Type IIa are what we call intermediate fibers. These are a blend between fast- and slow-twitch fibers.

An Android will not have any muscle as such most likely but consideration does need to be taken in the engineering of the motorized limbs to optimize them for the type of task they will be required to do.

Body Energy Considerations

A human being distributes fat cells around one's body based on the food we consume. The fat contains energy and also heat generation.

In an Android we typically see main power packs and secondary power packs. Unless an Android eats I personally see no reason not to stick with the traditional power pack approach. It might be useful to have some kind of built in solar technology ability where it makes sense.

Chapter 15 Goal Setting And Learning

Actionable Brain Cell Messaging And Enabling DNA Apps

According to this theory a Neuron can change its behavior based on the types of messaging that it accepts. In early development, a Cell is actionable based on the messaging that it receives.

Therefore we can have different states for a Cellular

Learning

Optimizing

Lookup

Often for an important decision people are asked to sleep on it to make sure that they can make up their minds completely.

One can imagine that the brain consolidates any decision while sleeping and receives some kind of messaging.

MESSAGE ACTION OPTIMIZE LONG TERM MEMORY

In this case the Neurons switch on Optimization DNA Apps.

```
MAP[
OPTIMIZE
ENABLE OPTIMIZER APPS
]
```

Therefore although the such brain is a fixed structure is behaves differently depending on the types of messaging it is receiving. A human being can therefore be seen as a Super Cell which is actionable based on different messaging such as Advertizing.

In the case of learning this can be a form of optimizing ones memories. General hunting strategies can be devised from specific examples. Therefore the brain needs to break down stories into their general components which are more commonly called tactics, general instructions and strategy. To deduce these one needs examples to work from like old Roman battles and positions of troops.

A baby spider for example will have general instructions like

Find the tallest object like grass

Climb to the highest point

Extend web into wind

The memory must be general enough and to do this according to this theory, specific examples need to be optimized and then stored in firmware memory if they are deemed to be related to high priority goals.

Hormones And Actionable Messaging Of Cells

How are Cells DNA Apps switched on and off if there are multiple cells? The answer according to this theory is that this is achieved by different Hormones in the blood and brain. The benefit of doing it this way is that it affects all cells in the general area.

An Android should also have System Wide Messaging as opposed to just per Actor Cells. This way it acts like a Multi-Cast Actionable Messaging Protocol.

The Hormone will do a specific task as I will show and change the state or operational mode of the cell for specific tasks.

Optimizing The Neural Net For Goals Using Neurotransmitters

Dopamine Neurotransmitters are located in the Striatum which is located beside the Thalamus while is where the SC is according to this theory.

The SC discovers content which matches goals such as EAT FOOD and then when these goals are matched, the brain needs to be optimized for these goals. For example we might smell food, the next step is to investigate which means getting up and looking. This requires action by the body in anticipation of finding something to eat which is a highly valued GOAL and needs to narrow its focus to achieve GOAL SUCCESS. Mice which are Dopamine deficient without this ability to narrow focus, when observed, they just barely eat or nibble food and may continually lose body weight. However if given L DOPA they will seek out the food and consume it. Without Dopamine they appear to be indifferent to eating.

According to this theory this is because their Cortex has not been optimized for this behavior so that the SC obtains GOAL FOCUS. Therefore Dopamine creates specific GOAL FOCUS in the brain by targeting Neurons related to specific goals and setting them into an excitatory state. Therefore the SC queue focus narrows on the goal in question e.g. EATING.

The two parts of the brain are included namely the DORSAL and VENTRAL system which controls movement and internal focus for tasks like eye focus within the rooms.

Without this focus the body do not rate the task highly and can be critical to the living organism. Also without Dopamine to help the brain focus one can have difficulty moving like Parkinson's disease because the brain SC cannot focus on movement. One can imagine the goal EAT to be a small percentage of the normal unexcited message but it must dominate when the task needs to be accomplished until it is over.

Also, the optimization of the Dopamine neurons in the Striatum must make their GOAL related connections during sleep periods so that the brain knows the GOAL relations. Therefore if one suffers a brain injury it is possible if the Neuron are present to relearn the GOAL related memories.

If one does not have these Dopamine connectors then a life form will appear lifeless and GOAL less.

An Android should also support this ability to focus using a Dopamine like GOAL related mechanism which ensures that the GOAL is achieved. The Dopamine substance should also switch on related apps and strengthen connections where required.

The Dopamine like Neurons have a tree like structure at the Axon level which is a form of multi-casting of the Dopamine substance. Also the receiving Neurons need to understand that this causes these Neurons to go into a HIGH FOCUS state.

DOPAMINE MESSAGES[

SET CONNECTED NEURONS TO HIGH FOCUS STATE

]

This may come in the forms of many signals to the SC. Also from an evolutionary viewpoint these types of messages make the subject feel good in anticipation of achieving the GOAL.

In the case where people fake this or other drugs to get an artificial high, the reward is lessened by the body if the GOAL is not actually achieved which is a safety mechanism by the body.

Using Dopamine To Learn

The Dopamine Neurons strengthen their connections when a goal is successful as part of a nightly optimization.

Memories flow through the system

learning=

SC messages

to working memory

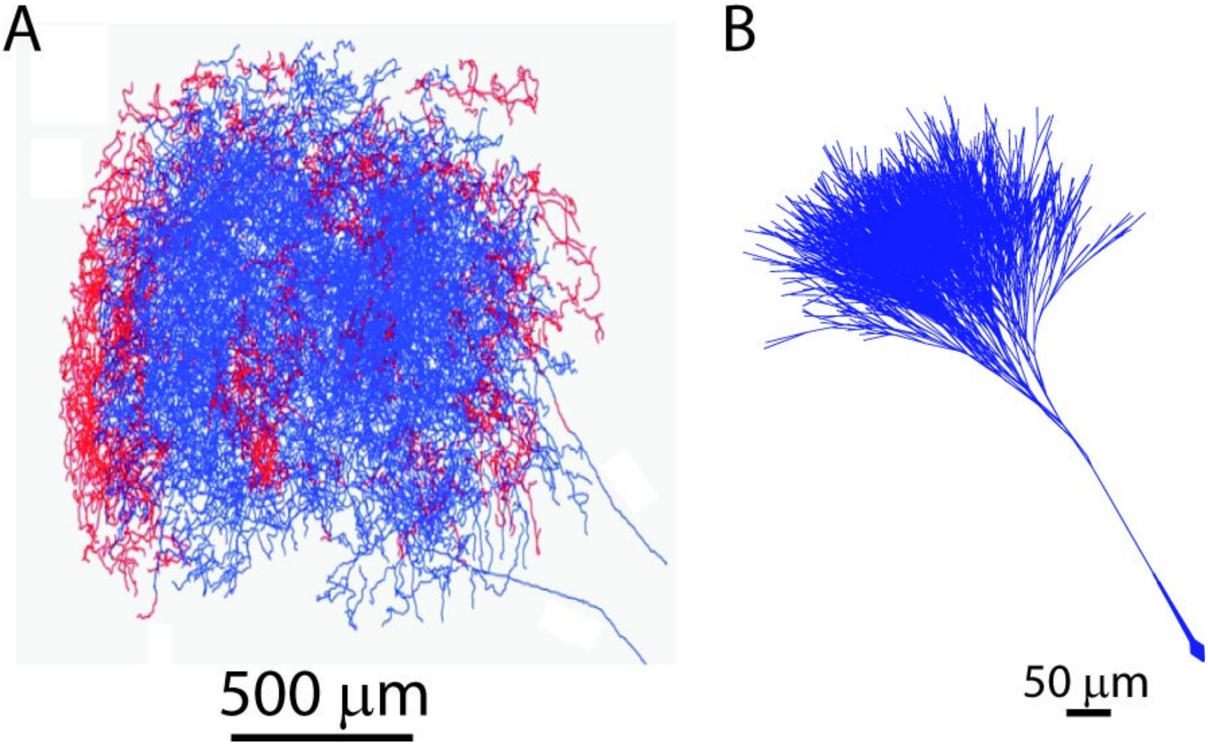
to daily memory

to long term memory

to (circadian) sleep optimization of neurons including Dopamine ones

and while awake using dopamine receptors during daily GOAL based tasks

A typical Dopamine receptor looks like this tree like structure where the GOAL maps to many related Neurons.



Figuring Out The Percentage of Dopamine

When a goal like EATING “could be” achieved, the SC needs to calculate how much Dopamine is required. To do this there needs to be a GOAL_CALCULATOR app in the Neurons which determines the chance of success. The algorithm needs to factor in

Current matching pattern

Previous success or otherwise

From this calculate a percentage

Ensure that the Dopamine receptor only produces this percentage to related Neurons

An Android should use this approach. Therefore the Android also needs to store where the success or otherwise of the approach in some kind of GOAL STORE is.

Melatonin And Deep Learning

When we sleep MELATONIN is produced in our brain. This signals that our Neurons should calculate Long Term Potentiation values between the Neurons. This is an Optimizer state that the Melatonin places the Neurons in.

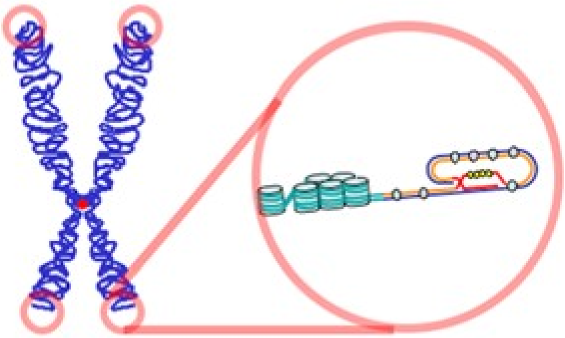
Therefore an Android can also have some kind of Circadian rhythm when it can do these types of learning optimizations.

Chapter 16 Evolution And Aging

Here I will discuss how a body evolves and how this is handled in this design. Part of this process involves understanding cell aging and the ability of Cellular life to dynamically adapt to a changing environment and encode new strategies for future generations.

Aging and Telomeres and Telomerase

Earlier I talked about WAIT and DONT_WAIT signals. How is this this encoded in DNA? Research has shown that at the end of each chromosome are telomeres which are needed for cell division. Each time a cell divides the telomeres get shorter. When they are spent then the cell will not replicate any more. Replenishment of the telomeres is achieved by telomeres. If one is subject to stress hormones or environmental factors the amount of telomerase drops and the cell replenishment shortens.



Therefore we have a dynamic system where stress has a direct effect on the cells and how long they replicate and replace one another/mitosis.

Therefore if we have a cell and is wants to multiply, the Telomeres define the life cycle of of the cells covering

GROW – when there are telomeres

DONT_GROW – when the telomeres are spent

Telomerase supports an ADD_GROW_COMMAND to Chromosome where it is required.

If the Cellular system is stressed for example in the case of smoking, the ADD_GROW_COMMAND is lessened during stress periods.

An Android therefore should have a stress response to an environment and should not try to heal itself while it is under attack and wait until the system is not stressed.

The Origin of Life – A Static Version

In this theory the origin of life is a pretty simple starting bio-organic program that has a single goal, one app and some internal and external messaging.

GOAL ENGINE:

REPLICATE (EXECUTE COPY MESSAGE)

APP:

COPY (COPIES ITSELF THEN SENDS HELLO WORLD MESSAGE)

MESSAGE:

INTERNAL : COPY MESSAGE CALLS COPY APP

CAN SEND : HELLO WORLD MESSAGE

CAN RECEIVE : HELLO WORLD

That's it. The idea is that this life which has just evolved. This is obtained using natural selection and the Darwinian ideas. The other idea is that the GOALS, APPS and MESSAGES have grown over time using self-improvement messaging. I'll talk some more about this later.

The idea is therefore that we have an organism that copies itself and they can all say HELLO WORLD to each other.

So what does this look like? This is like a primitive life form filling a pool of fluid for example in a blind fashion and messaging each other.

They are all saying HELLO WORLD to each other using some mechanism.

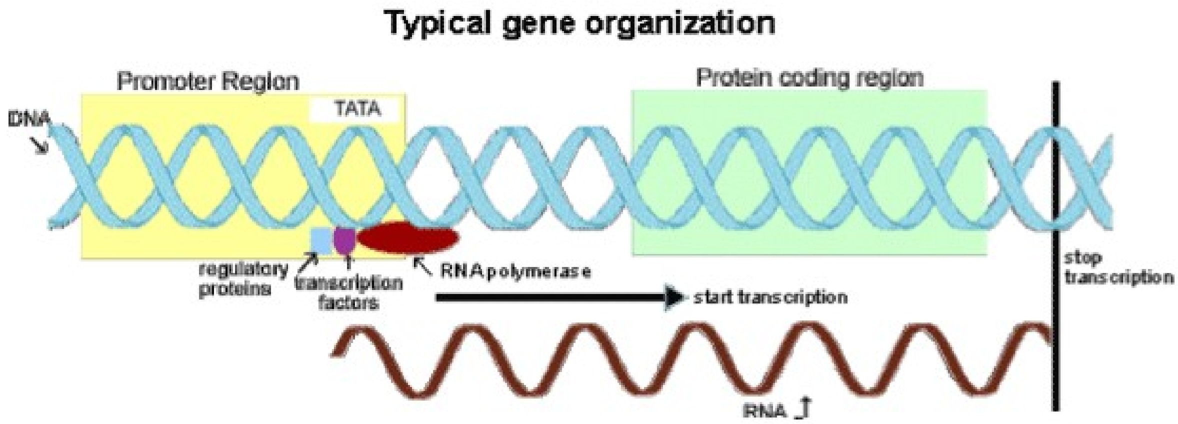
The GOAL engine does not call the APPS directly, it is the job of the GOAL engine to fire messages which are in turn converted into APPS.

An Android should have GOALS, APPS and MESSAGES.

The problem with this approach is that is it too static and cannot evolve. So a more intelligent approach is required. We need to come up with a Dynamic Design for this application so that it has learning goals and is aware when they are reached and not reached and subsequently adapt. This is for later but the idea of Apps, Goals and Messages still holds.

Handling Messaging, Apps using Promoters

How does DNA handle an incoming message and know which DNA App[s] to run? Research has shown that DNA has what is called a Promoter which is a fragment of DNA which matches / resides beside a particular protein message. From this, the App can be transcribed and then ran so that it can handle its function.



An Android Actor should contain some thing like DNA so that we can have some kind of adaption mechanism which I will go into later.

Cells Containing Memories and Remembering and Communicating

Most people assume that we need a brain to remember things. However, it has been shown that plants can remember things even through they have no brain or nervous system? How does this work according to this theory?

The types of things a plant can remember is the direction of up if it is made to lie on its side. Plants can also dance to music. Plants also find and cling onto things.

The wat this works is that state information is stored in the Cytosol and this records which DNA Apps are switched on and any runtime state and messaging.

```
RUNTIME STATE
DNA APP A ON
APP A STATE B = TRUE
APP B STATE REORIENT LEAF = TRUE
END RUNTIME STATE
```

In a living organism these are protein markers which represent the runtime state of the cell. In a Types Actor we need a runtime mechanism to handle this state information. Therefore it makes sense that a Typed Actor keeps the state information for a fixed amount of time so that it remembers. Some information should last longer than others.

The information can then be shared with other plant cells via the Intracell communication mechanisms.

```
INTRACELL COMMUNICATE WITH SPECIES
INTRACELL COMMUNICATE WITH NON SPECIES
```

Trees for example which are predated upon too heavily will warn others of its species and also undergo stress and emit defense messages, like producing high degrees of tannin for animals that eat their leaves.

INTERCELL LEAVES EATEN STRESS

PRODUCE TANNIN INCREASE MESSAGES

An Android should employ defensive measures if it undergoing stress.

Dynamic Defense Strategies

Clearly a life form which is based on a static program where there are multiple predators will fail which is currently how software engineering works for the most part. Therefore one needs a dynamic defense strategy which deals in real time with immediate threats and longer term defense where one passes on DNA apps.

So we move towards a model where there is a sophisticated threat analysis messaging e.g. a caterpillar is eating leaves which is detected by the saliva content and a variety of survival strategies are implemented e.g. emit chemical to attract predator of caterpillar.

THREAT ANALYSIS

CATERPILLAR THREAT MESSAGES

INSECT ATTACK MESSAGES

…

CATERPILLAR THREAT INVOKES APP ATTRACT CATERPILLAR PREDATOR

The RUNTIME state is also part of the calculation. For example, if the predator does not arrive, the plan will WARN OTHERS.

The creation of a new type of App is something that will be talked about later.

A dramatic option is to change the type of flowers a plant has, for example a tobacco plant can change from a moth pollinator to a hummingbird pollinator. This requires switching on a different type of App

DISABLE MOTH POLLINATOR

ENABLE HUMMINGBIRD POLLINATOR

The idea here is that plants and all life have survival strategies which are in place and will only be enabled in certain circumstances. This requires that the App for this is already encoded.

Moving forward, I will propose a mechanism for all Message App interactions that check to see their success or otherwise so that the organism in question knows if it is succeeding or not and from this one activates stress threat messaging.

An Android should adopt this.

Measuring The Success or Otherwise of a Message based App and Message Chains

The general model of a Message activated App is as follows.

An App is activated by a promoter Message.

MSG-APP

The App needs to know if it has completed its task successfully and requires a validation state, This is in the form of Validation Messaging.

MSG-APP-VALIDATE

It's important that all Apps report on their state once their task is done so that the process can proceed.

Also one Message can activate more than one App to Express its Genes.

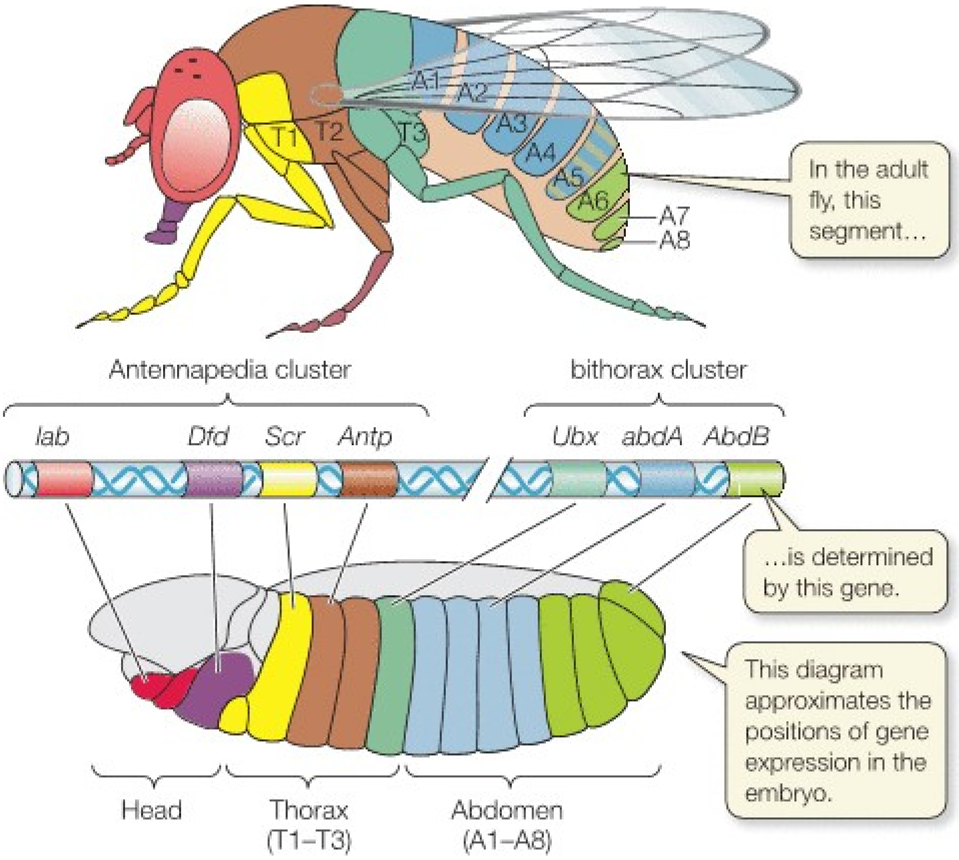
For example, an App might be required to create a Dynein robot motor. It there are not enough raw materials, then a RESOURCE_LOW message is generated otherwise the next step in the Message Chain is implemented.

In traditional Software Engineering software has a fixed structure where an engineer can read the start and middle and end of the program. In this model, **Message Chains** control the flow of the operations and what the steps are. This is like Message Stories but where there are Chains of operations where each reports success or otherwise.

The advantage of this approach is that if one encounters a problem the whole program is not held up. Over time the organism can then evolve new messaging to solve the problem reported and the Apps which are working proceed as normally. Typically in people a person with a genetic illness can continue to function with some disability but the MSG-APPS which function are separate from the problem because Messaging controls the flow. The downside of this approach is how to define the messaging and traditional software engineers may not be terribly comfortable with this approach but the upside is scalability and fault tolerance. As I will show we should be able to also evolve this system in the same way that a human and life does with some kind of drag'n'drop approach to Msg-Apps.

PAX Genes Implementing Generic Body Building Strategies

How can we design and build a being based on DNA Apps? Research has shown that individual HOX genes control various body parts (PAX for the eyes) so we can have an App per body part and controls its life cycle. So when we see one entire body this can be broken into dedicated HOX Genes.


A1
A2
A3
A4
A5
A6
A7
A8
T1
T2
T3
In the adult fly, this segment…
Antennapedia cluster
bithorax cluster
lab
Dfd
Scr
Antp
Ubx
abdA
AbdB
…is determined by this gene.
This diagram approximates the positions of gene expression in the embryo.
Head
Thorax (T1–T3)
Abdomen (A1–A8)

Research has also shown that the **same genes** control different body shapes. So the idea is to create a high level generic app for each body part. There are tables of HOX genes which map to different species. Many HOX genes are shared. So we can begin to see DNA as a mechanism for building **polymorphic lifeforms** of different shapes and sizes dedicated to different tasks.

Interestingly, the eye PAX gene in a fruit fly also works in a mouse even through we might be inclined to think of them as different. Therefore life is componentized and is pluggable into different life forms with encoded positional information within the DNA Apps.

Therefore we can see a design pattern where we can reuse APPS dedicated to different body parts in various body types where the requirements of the living being are slighting different.

EYE APP

CHEST APP

LEG APP

WING APP

and so on. Each one supports Messaging which covers its growth/assembly life cycle and may look quite different but uses the same Apps where it makes sense.

microRNA acting as Orchestra Apps

All Apps do not just build things. In real life when a building is constructed there is some kind of Architect and Site Manager who keeps track of everything and is aware of the detailed and correct steps for the task.

How does DNA doe this? Recent research has show that microRNA has the job of switching on and off other apps and co-ordinating them. Earlier I called these Orchestrator apps and by this I mean Apps that co-ordinate other Apps. So when a body is being built or handling complex operations, this is the job of these small pieces of RNA.

Other tasks for RNA cover building protein machines but research has shown this is about two percent versus ninety eight percent other tasks such as this. To a certain extent this is the business/goal logic of the DNA where it is not merely manufacturing things but intelligently building and controlling something according to a blue print. This ties in with the idea of a being knowing what its Physical and Virtual form looks like and how a vision system could theoretically work.

This is probably one of the most complex aspect of programming in the system and would require some time to figure this out in an Android design as to the most efficient steps to achieve the design. Logically this is where a trial and error model could be used in some kind of Evolutionary Adaption model for the shape and form aspects of the DNA as opposed to its Core App design in relation to various body parts. In the case of the Android we can consider the Orchestrator App in relation to some design team but in many ways this is no different as all product design typically is a trial and error process and so all products evolve in their own way even by human hands.

```
ORCHESTRATOR APP BUILD CHEST
SWITCH ON A,B,C
SWITCH OFF D,E
WAIT UNTIL COMPLETELY DONE
SWITCH ON W,X
etc;
```

Reverse transcription and storing code fragments back into the DNA

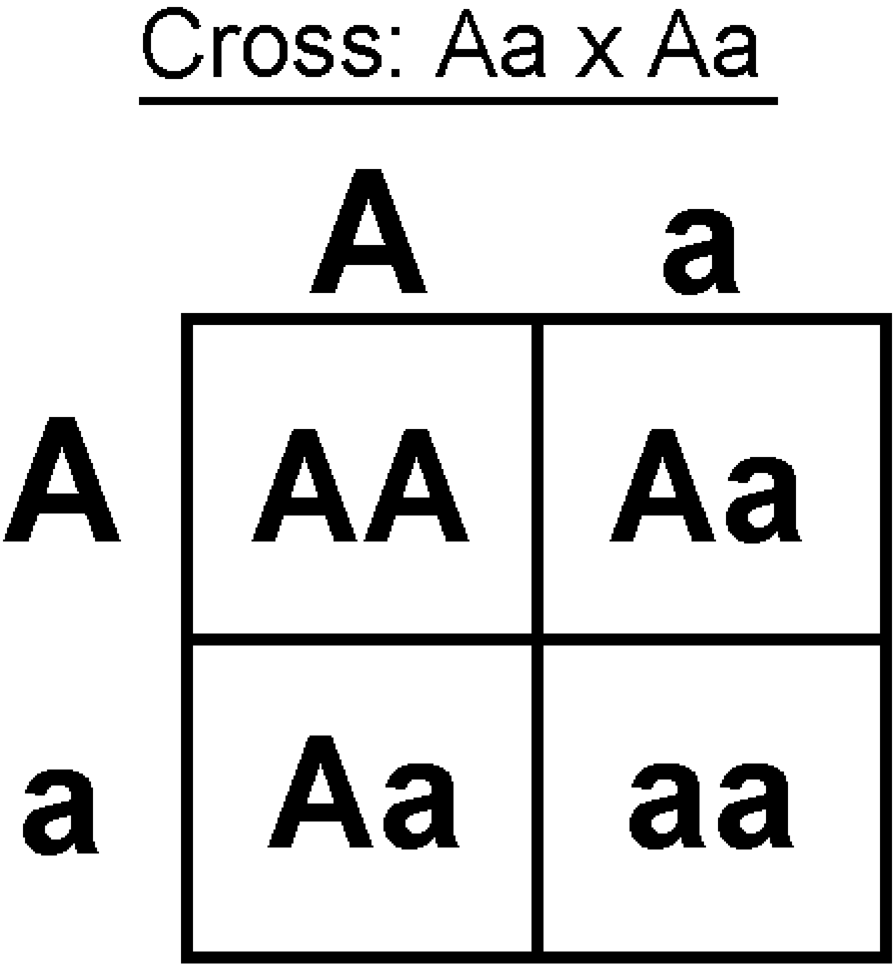
As a simple start to supporting the idea of a system evolving and being dynamic, one of the features that nature offers is called Reverse Transcription. In its simplest terms this is when a fragment of a running App can be re-encoded into the DNA Apps of the Typed Actor. Therefore we imagine a piece of external useful code being added to a live run time system and being made part of its genetic make up. To support this idea is extremely powerful and makes the Cell a much more dynamic system which interacts with its environment instead of having static programs.

```
REVERSE TRANSCRIBE APP FRAGMENT A
```

This is a large topic and this is a simple introduction to it at present. This is a must have feature for an Android but is very complex and covers security issues relating to the Typed Actor so one must also determine the cost benefit of said transcribed Apps. I will talk more about this later in terms of Bacteria and Viruses. As well all know some are good for us and some are very bad for us.

Heredity and Traits

When Mammals mate they have two possible combinations which were defined as Dominant and Recessive. Using a Punnet Square we can determine the likely-hood of the result. This idea ensures that each person is reasonably unique and humans are not clones. Therefore it makes sense that an Android design should possibly support this idea where each Android is not identical. So one can imagine them being the same but slightly different, for example in eye color and so on.



Cross: Aa x Aa

	A	a
A	AA	Aa
a	Aa	aa

It make sense that certain traits can be attributed to certain androids also depending on their uses. Also when humans interact with Androids they may prefer different physical traits and so on.

Heredity = App options

Traits = App results

Bacteria And Cell Co-operation

There are more Bacteria in a human body than there are human cells. Here we move towards a model of life and for an Android where the being is not solely self-contained and has helpers which communicate with it. If their interaction is to the benefit of the organism they can ultimately become a part of it and depend on one another. In the case of the human gut we have many beneficial bacteria there helping us digest out food and in return we provide a home for them. Therefore we can imagine an Android which employs a similar co-operative mechanism which keep it on track for its own goals.

Sample External Apps/Machine for

Cleaning

Preparing Recharging Stations

etc;

An Android can therefore see a human as part of this co-operative mechanism.

The Android helps the human and the human helps the Android.

Viruses As Code Factories For Genes

This is a big topic but one of the major questions around evolution is how did genes evolve because they contain a lot of genetic code. The current answer is that they evolved gradually but software engineers know that chunks of code need to be written and not just one line at a time in some cases. According to this theory, the larger gene changes are designed, tested and employed by Viruses. **In this theory Viruses are seen as Coding Factories for living organisms.** Not all code is good for the organism though. Not all evolved genetic code is employed by genes but in this theory this is one of the Engines of Creation namely Viruses. This will be covered in more detail as I move forward but for now this is the concept.

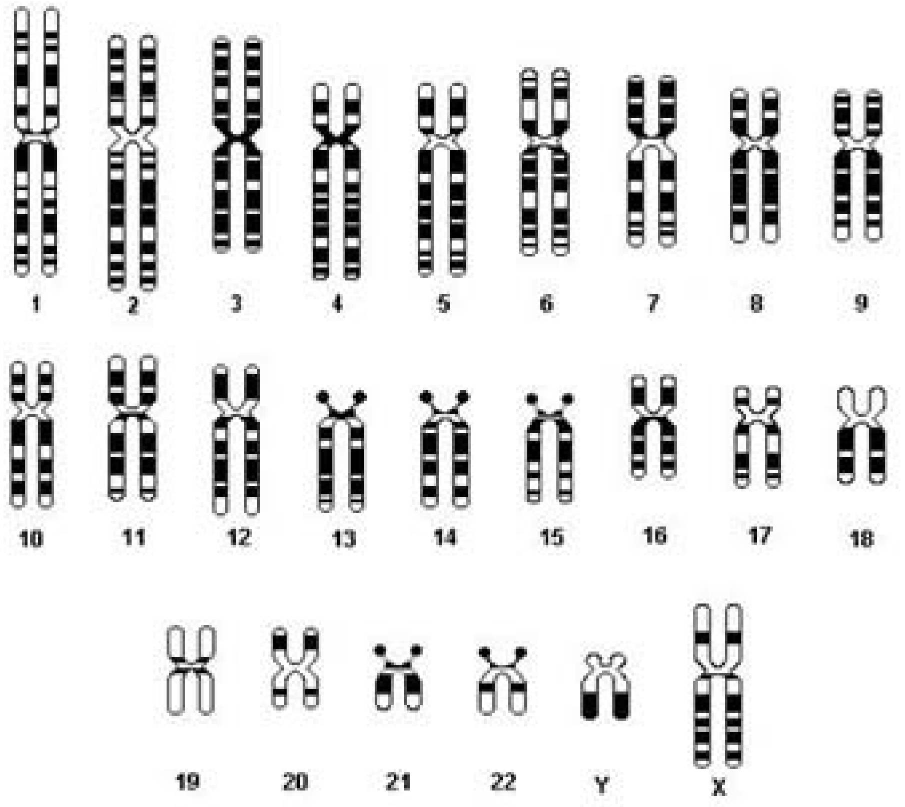
Virus = Code Factory Snippet Generator

Using this approach we can imagine that there are Android code snippet generators. Using a modern analogue we can provide interface points for Android where we can code additional behaviors and add-ons to customize them. This idea is already popular and can be readily applied to an Android design.

Later I will take a look at the various virus types and see what kind of features an Android could employ with these from a design viewpoint.

Being a Msg-App on one chromosome App-Package versus another and timing

Each chromosome is seen as an **App Package** in this theory, similar to packages in certain software languages. They are separate and distinct in the nucleus and each Chromosome does a specific job.

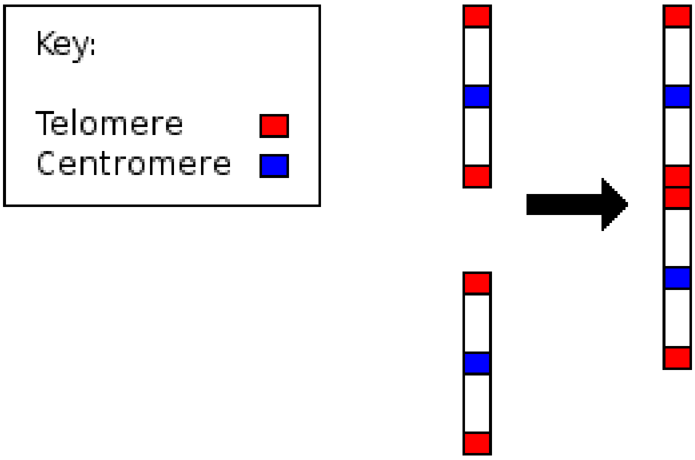


They vary in size and function and are numbered. Each houses different number of Msg-Apps. We can then ask the question what is the benefit of being on one Chromosome or another? In other lifeforms there are no Chromosomes so the major reason according to this theory is **timing.** Related Msg-Apps are co-located on the same Chromosome. Also each Chromosome has the equivalent of a timer mechanism at its end which is part of the Telomere structure. So if we want to co-ordinate activities it makes some sense to have them on the same Chromosome but I am sure there are some exceptions. We can treat this as a general rule for this theory.

Therefore an Android should employ this design pattern. Later I'll talk about the various jobs of the Chromosomes.

Evolving Chromosome 2 and 3 into one Chromosome to assist human evolution

In hotly debated Intelligent Design versus Evolution, it was argued that the merging of Chromosome 2 and 3 into Chromosome 2 which gave us human DNA as opposed to Chimpanzee DNA which has 24 versus 23 Chromosomes proved evolution. This makes sense but it's unclear to me as yet the mechanism of this merging but it is a moot point. It clearly happened.



This folding of two Chromosomes into one is an interesting case because using it we can take a look at how this could help make us more intelligent. Basically we are placing two sets of Msg-Apps onto one Chromosome. Let's examine how this could make us more intelligent according to this theory. For the first point we are **merging and co-ordinating** Msg-Apps (which previously had not been in a Chimpanzee) because they share the same timings and location for incoming messages. Recall the section on the human brain where I argued in favor of a human brain being able to combine messages to achieve an increased understanding of the information e.g. sound and vision (as a basic), vision and semantic information (human), vision and music (human), motor function and music (human) etc;

Therefore a Chromosome like this would have more Synesthesia where it can process more messages in parallel e.g. combine HOSL messages with for example the Vocal messaging and with this we can produce human speech and thought.

Merging Synesthetic Messages On A Single Chromosome To Achieve Higher Intelligence

If we take a look at Chromosome 2 we can see that it manifests certain behaviors when there are defects giving us a clue to its behavior. These defects give us an insight into how the Chromosome behaves. According to this theory, this Chromosome should be merging messaging to help us obtain higher intelligence and be related to the brain generally. Is this the case?

Let's take a look at well known Human Chromosome 2 issues which show this is the case. This is **tentative** proof as opposed to absolute proof.

Autism : I have already shown that this is related to difficulties with message processing in the brain.

Synesthesia: This is an important one because it shows that messages are being combined in unexpected ways which I've covered before and I've already argued that the reason for humans ability to speak is their ability to connect their HOSL to the Vocal Messaging which is a standard synesthetic feature in healthy humans e.g. our ability to talk or verbalize our HOSL.

Also by being on the same Chromosome things happen faster and are synchronized which I will cover next; namely the speed and co-ordination at which things happen.

Plants Versus People And Speed And Reaction Times

Plants are know to have many more genes than humans. One would imagine that this should be the other way around. However a key difference between humans and plants is that plants do not need to move as fast as humans or life that moves over a Safari plain for example or inside water.

Therefore we can see that although plants do move and sense their environment they do so more slowly according to this theory because they have so many Chromosomes and the inter-Chromosome messaging is handled more slowly. An Ant for example has only one Chromosome and can move quite quickly and handles limited messaging as I've shown.

So the rule of thumb is:

If you want fast Msg-App reaction times between them place them on the same Chromosome.

If you want to combine and co-ordinate messaging place them on the same Chromosome.

Virus Species Jumping And Testing DNA in Avian Flu

The avian bird flu is a good example of destructive Apps gaining access to a Cell. The H1N1 refers to the H1 key and the N1 gene which in our case is a Msg-App.

Therefore it's important to have good security to the typed Actor in terms of which Apps are allowed entry to the cell. Here we enter the world of Security which is a large topic.

Sufficient to say a cell needs to have a **Security Manager** which checks the Keys of entering Apps.

SEURITY MANAGER CHECK APP KEYS AGAINST CELL KEYS

In a protein world this is a protein signature.

If the app passes and is flawed then this can be fatal to the being in the case of the avian flu virus.

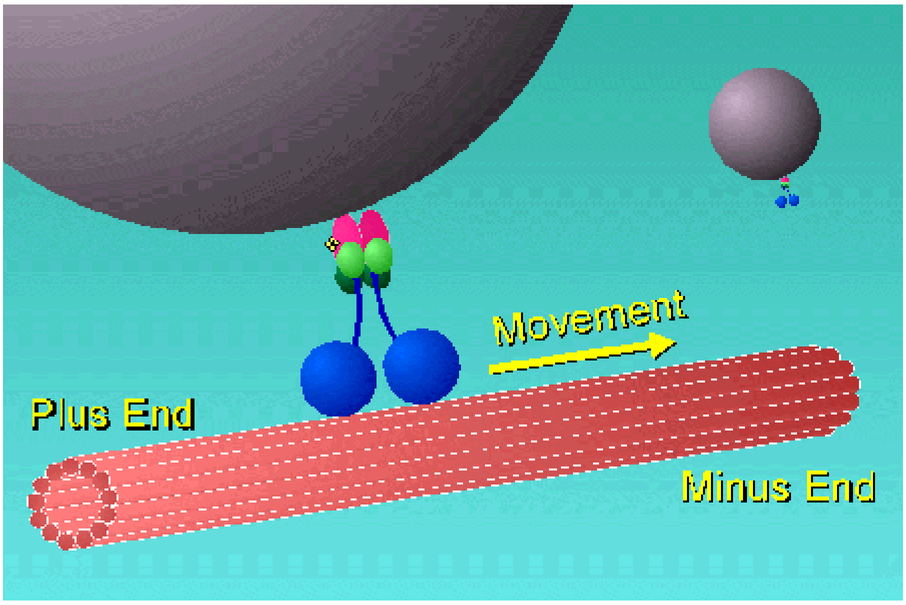
However there must be some viruses and in turn Msg-Apps which enter the organisms DNA and are not fatal.

Most viruses in the real world as we see tend to cause trouble are the ones which we are concerned with by some Viral Vector techniques attempt to add medical gene re-engineering.

An Android should have a Security manager for Msg-Apps wanting to be added to an App-Package.

Miniature Robot Helpers And Dynein creating Helpers And Connections

Most DNA builds miniature protein robots. One of these is Dynein but there are others. These make sure that cells function correctly ferrying things from A to B and so on. In the world of software engineering we can think of these are data structures that are ferried through the code.



So we can map the idea of Robot Helpers with Software Helpers

CREATE HELPER E FERRY A FROM B TO C USING CONNECTION D

An Android should support the idea of Helpers.

Also they can contain connections.

CREATE CONNECTION D FROM F TO G

This should also be supported by the Android.

The Number of Apps / Genes On Chromosomes And Contexts

There are between 22,000 to 25,000 genes in human DNA which are distributed over our Chromosomes. So for example, one Chromosome may contain approximately 11,000 Apps.

The human body is reasonably complex so it is safe to assume that these genes have various input factors when they run which means they operate in different Contexts.

For example a stem cell can become a Liver Cell or a Neuron. Therefore, each Cells runs within certain Contexts.

Therefore we need an evolving Context Manager.

```
CONTEXT MANAGER
LIVER CELL
NEURON CELL
…
END CONTEXT MANAGER
```

We also need messages that Change Context

CHANGE CONTEXT TO LIVER CELL FROM STEM CELL

Transcription Factors And Parameter Passing To Apps And Message Lego

The Promoter Region of an App does not just take a Single Message to activate it. This is too simple. It has been shown that human beings have dozens of Transcription Factors which combine in a form of Message Lego where one Message Connects to another and is context driven. Therefore we have a Promoter Manager which understands all the messages needed to activate an app in various ways.

```
PROMOTER MANAGER
CONTEXT LIVER CELL
REQUIRES
MESSAGE A,B,C,D
END PROMOTER MANAGER
```

This is equivalent to calling an App with certain input parameters in order to start it.

Deactivating Apps

Here we can extend the PROMOTER MANAGER to not activate with certain inputs?

```
MESSAGE CANCEL LIVER CELL APPLICATION
```

This is like an App having a flag which is

```
disableApp = TRUE
```

Enzymes As Evolving Molecular Engines Of Change And Transformation

Enzymes run along the DNA and do the Transcription of the App/Gene when the Transcription factors are correct and create the RNA strand.

The Enzymes are seen as Engines in this theory which facilitate different operations and their molecular structure. An Engine transforms structures or creates new ones.

ENGINE WRITE RNA CALLED BY TRANSCRIPTION FACTORS A

Here the Enzyme has the possibility to understand the Transcription factors and write the RNA fragment knowing the Context in which it is doing this. According to this theory, this can lead to different results possibly.

The Selfish Gene And Genetic Determinism

An idea popularized in the 1980s and later discussed is the idea of why an insect that would not breed would self-sacrifice itself for the nest. The idea was that the genes in the insect were shared by the leader of the colony and the Gene itself is Selfish. This is also called Generic Determinism and is a very narrow idea now that one sees how dynamic Cellular life is but the idea is worthy of being considered.

According to this theory life communicates with its own type such as is the case of plants when they are attacked and also requests help from other predators which it is not related to. Therefore, one can see all life as serving the Msg-App to either succeed or fail based on its goals.

An Android could therefore help save a human even though it might jeopardize its own life not just because it has the programming to do so but because humans are beneficial to Androids in general. The Android does not need just genes to do this. It does this because humans help obtain its goals of improving. Therefore humans and Androids should share common goals that are mutually beneficial e.g. the Mars Rover is a good example where the Rover benefits by human support and the humans benefit from its work. Overt time Humans will evolve the design of these Rovers because they are useful to humans but the Rover benefits from this as well.

It is therefore not Selfish (which is more of a one way transaction) but Mutual Co-Operation (which works both ways).

Therefore one could imagine a Core Program that started life on Earth with a simple goal to simply replicate. All the different lifeforms we see which include this goal are serving that basic need in a form of mutual co-operation.

A good example also of evolving mutual co-operation is a Wolf which initially hunts and eats humans to survive in opposition to the human goals to not be eaten. However if the Wolf is tamed and becomes a safe pet it will use its keen hunting skills to protect the home of the human. In return it gets a regular meal and is cared for. Therefore the goals are matched and the relationship is mutually beneficial.

These are the goals that must be striven for in an Android.

Building and Growing Bone That Detects Gravity and Weight

A body is a dynamic thing unlike the more static Darwinian idea in that the shape and form is constantly detecting the environment independent of reproduction. We are familiar with a body builder but also importantly our bones also detect the weight and the gravity we are subjected to. Astronauts in space find that their bones decalcify in zero gravity because body sensors realize that not so much Calcium is required. Therefore they need to do constant exercise to keep body strength.

In the case of an Android that say would visit other worlds this design idea makes a lot of sense because if we design an Android for one strength of Gravity then it might not work on another work like say Venus.

On can imagine joints which form a lattice like a human having endonchrondral bone formation for length and strength and appositional growth for width. Internally a lattice would provide centers for ossification so that it can add or remove metal strength in the limbs. Clearly one needs less energy if one is lighter.

Chapter 17 DNA Programming, The Kernel and Genetic Algorithms

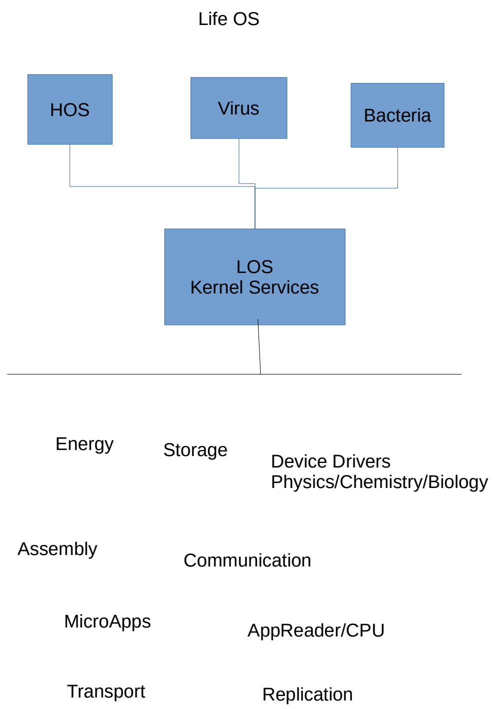
Defining A Kernel and LOS

Clearly there is much more life than just human life and the mechanism humans use are shared by all kinds of other life on Earth like Bacteria and so on. Therefore we can extend the architecture to include a Life OS which sits on top of the LOS Kernel. The Kernel provides the core services which are used by all life.

We can therefore think of human life or HOS life as an application within the LOS Kernel space.

The Kernel Services cover basics services such as Energy, Storage, Device Drivers for Physics, Chemistry and Biology, Assembly, Communication, MicroApps, AppReader/CPU, Transport and Replication.

An Android should have a Kernel to support services like this. In a typical electronic device the Kernel covers the list of devices attached and so on and provides disk space to store apps.



Genetic Algorithms And Epigenetic ones

Here I will discuss how we can treat DNA as a programming language and apply the concept of Genetic Algorithms.

A genetic algorithm uses a goal as the target and measures the fitness of the DNA Apps which provides the features for the goal e.g. feed successfully on the insects or build a snake that flies and is not allergic to poisonous insect is the goal but the features are randomly distributed. The solution combination is distributed across different beings of the same species which mate and using random mutation splitting we try to find the best fit for the goal combined with mating. This leaves a large solution space and an indeterminate number of matings to get to the best result.

If we implement something like this in an Android, the key is to keep the genetic algorithm solution space wide enough to maintain diversity e.g. we do not design a genetic algorithm wipe out crocodiles because they are part of our ecosystem even through using a highly optimized solution space this might seem like the mathematically optimal solution and because we think they are dangerous. As humans, we have learned to make these distinctions over time. The algorithms applied needs to have this degree of sophistication. They need to be code reviewed and get regulatory approval.

Note that using an Epigenetic model, one does not need to breed a new instance in order to solve for certain goals. The genetic mechanism leads to a 'Goal Based Epigenetic/Dynamic Genetic Algorithm' where using a control system one can switch on/off Apps/Genes to achieve goal. **Evolve the genes/app dynamically for certain cases and test for fitness in relation to the goal (rather than mate).**

Goal Based Epigenetic/Dynamic Genetic Algorithm

In this case, we do not need to breed over and over again to meet the goal. We apply a Goal Based mechanism which switches features on and off until we solve for the solution in question. There is a possible solution with the embedded Genes/Apps. In the case of a snake needing wings and not being poisoned by an insect this does not apply but rather to the idea where we can apply different solution strategies which are encoded as apps for problem solving.

LOOK FOR OBJECT A

SWITCH ON APP A

FAILURE

SWITCH OFF APP A

SWITCH ON APP B

SUCCESS!

The Twin Pillar Approach to Life aka II – Reaching Nash Equilibrium And Evolution

How is it that life evolved? A famous gaming mechanism one can use is called Nash Equilibrium. The idea is that if one is at a set of traffic lights and there are no police or laws against breaking the lights people will still use this mechanism because it makes sense to use it. So we think of the system and items within that system operating in Equilibrium with one another.

Let's apply this idea to the idea of HOS and LOS. LOS runs the system and HOS operates within the system. Therefore we can think of a living being as in Equilibrium with the LOS. Both the LOS and the HOS get something mutually beneficial out of the relationship. The LOS gets to replicate itself within a living being and the HOS (or another other kind of specialized life) gets to make use of the LOS in order to specialize. This is a system which is in Equilibrium aka Mutually Beneficial.

So using Gaming Theory, we can model life as a '(LOS) self-replicating versus (HOS) self-organizing NASH equilibrium'

So a living being serves two main purposes

1. Self-replication home for LOS

2. Self-organizing home for a specialization e.g. HOS Human.

Therefore we can think of this as a twin pillar approach to life aka II which stands for Self-replicating and Self-Organizing co-operating. Each contains some way to measure fitness for purpose and uses the Messaging system to inter-communicate with each other.

Therefore humans look out for themselves which is the HOS/Genes and importantly they also look out for each other which is the LOS/Life itself and other versions of themselves.

In nature, we see species predated on protect themselves but also protect the herd by messaging them.

Therefore Androids should be programmed to look out for themselves but also for Androids in general and also humans who created their LOS. For Androids or machine life to be useful for the LOS/HOS, there needs to be mutual co-dependence. The LOS can be in the form of any life the machines contain. This could even include Bacteria for panspemia! For machines to be useful to HOS, there are many uses where humans can use machines in many useful ways medically and or other ways which can help one with specialized help. There are too many to list here.

Therefore Evolution can be seen as:

Evolution of life is the point at which LOS self-replication reaches Nash Equilibrium via messaging with self-specialization to form a new life e.g. HOS.

This process is ongoing within all life. The development of machines by human life is a form of self-specialization in this process. The machine Kernel is the LOS.

How Molecules Speak To One Another With Message Sets And Set Operators

Cells talk to one another with Messaging. In this theory, the idea is that these messages operate on the Apps with Set Theory to generate Logic and Reasoning. This is a big area of interest in pure Math. An app can be seen as a message set. The idea here is that the Message Sets are complex and both the LOS and the HOS can compare result sets and determine fitness of purpose. Logic and Reason can be deduced from the Message Sets and the Operators assist in this idea.

LOS result set

HOS result set

Compare two result sets for Nash Equilibrium

For example one does not need count an infinite set to need to know if they are different but just find one pair which is different. One can imagine that the set of entries are molecules and can be compared by some kind of Set Operators

LOS Set Operator Compare [Hos Set, Los Set]

These Operators are like the readers for DNA.

Therefore DNA itself can be viewed as a set which is Operated upon by a HOS Set Operator.

HOS Set Operator Read [HOS Set] ← DNA Reader

There can also be Set operators to store sets can create DNA and also operators to copy the sets.

We can visualize this as Mytosis.

So LOS Engines can also be viewed as Set Operators.

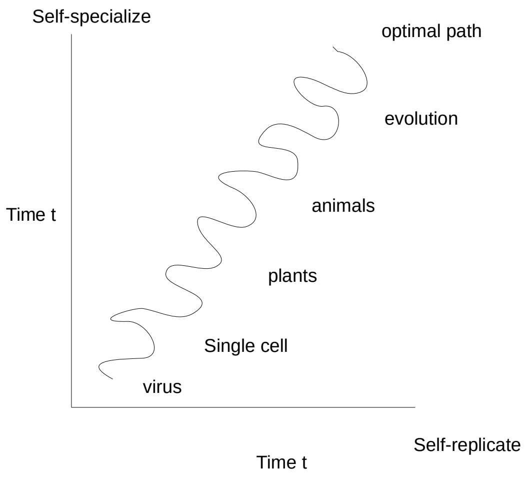
Therefore what we call Life can operate on Set Theory with Operators.

An Android should operate using this type of approach.

Evolution As Two Competing Strategies

We can see life evolving from the need to spread out but also specialize. If an animal becomes too large and is not specialized enough then it will fail from an evolutionary viewpoint. In the case of the dinosaurs we can imagine that they succeeded on self-replication but not becoming more intelligent over their time on Earth. Panda Bears on the other hand have very specialized needs and are adapted to their environment but fail on mating / self-replication.

We can draw this like so.



An Android needs to be useful at something in order to make itself useful in general and then it will be worked on and improved. Otherwise we can end up with esoteric designs that may not be that useful. Early home computers and early bikes and cars are a good example. Over time the design has become obvious.

Even the Universe adheres to self-replicate / self-specialize. This is what the Big Bang is so this looks like a general framework for our reality.

Engines of Creation Are Operators

Using this design pattern we can see the engines that zip around our body akin to math set operators.

It's all about the operators! They operate on sets of data. They can also measure time and the rate at which messages arrive. They can also determine strategies and next moves. They can build and assemble and the operators can evolve.

The operators can contain complex logic and interact with the sets of molecules they need to. The Android should contain operators like this.

The First Operators

If one imagines that the first signs of life were operators that implement the proposed design pattern, then they must support at a minimum

1. Replication
2. Refinement

So they had the ability to copy themselves and copy other sets of molecules.

They also had the ability to make similar copies which were slightly different and this in theory kick starts contemporary evolution.

The list of refinements need to also be compatible with replication so they are in equilibrium so operators must theoretically always support replication. In order that an operator can support replication then we can imagine that the right food source is available for replication. Therefore refinement is seen as an aide to replication but they are co-dependent.

Each operator will also need an energy source to drive it. In modern biology this is the energy release from ATP to ADP.

Operators can contain other operators and operators can be very complex. An example of a large operator according to this theory is a human Cell. This contains other smaller operators and can replicate and is refined into other forms. In these evolved cases the operators communicate with one another.

We can model operators like this in an Android.

Operators are analogous to Actors but based on set theory.

Operators Operating On Sets

So we can think of an Operator which can have independent movement and contains an energy source. It takes in sets of data and sends out sets of data. It has the ability to replicate itself and it can also refine the set data. Operators can be very large. For example a person or an animal can be an operator but also a single cell can be an operator. A dynin motor inside a cell is an operator. However, the DNA is not an Operator. It is the set data that the operators operate on.

Let's take some examples

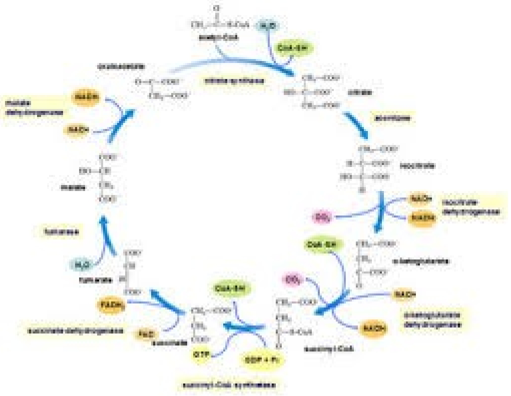
Operator	Refines	Replicates	Is A Set
Person	Yes	Yes	Nos
Female	Yes	Yes – species level	No
Male	Yes	No – species level, needs female	No
DNA	No	No	Yes
Cell	Yes	Yes - Mytosis	No
RNA	No	No	Yes
RNA Transcription	Yes	Yes	No

Operators Chained Together

Operators take a set input and then produce an output set. Operators know their inputs and know their outputs.

A good example of operators chained together to produce some beneficial result for people is repiration which takes in oxygen and produces energy.

This involves many operators transforming inputs to more useful outputs.

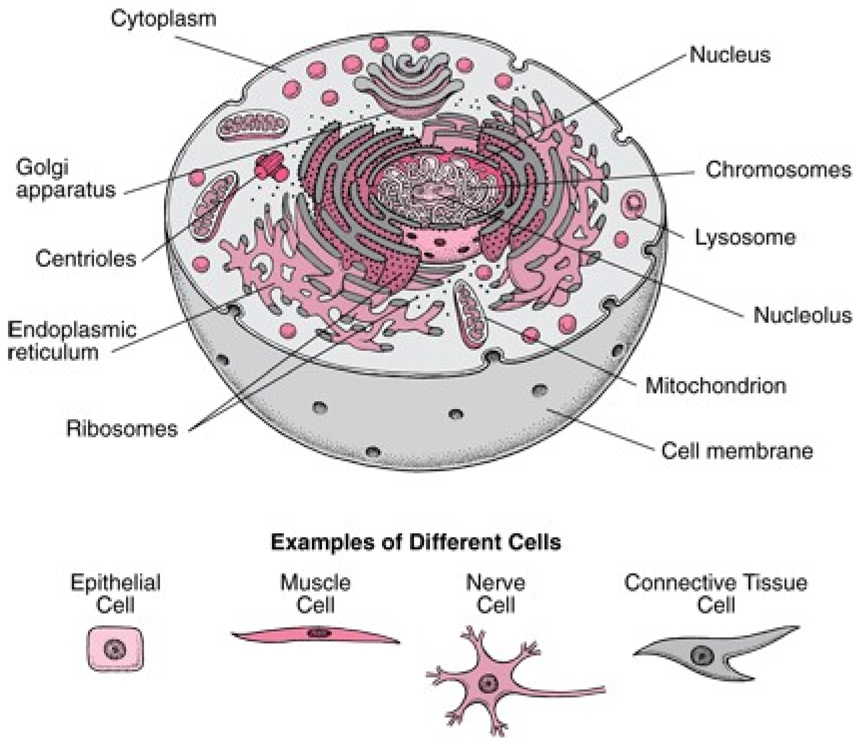


 Each transformation is an operator.

Operators inside other operators

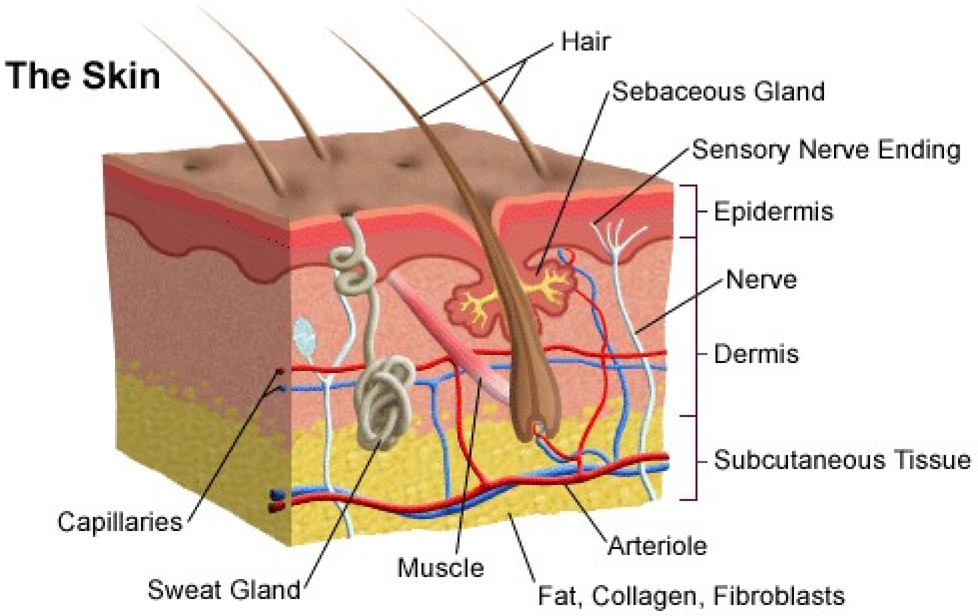
We can imagine a design pattern which builds out and then up. The larger operator takes set inputs and has outputs. Inside the operator are other smaller operators which do specific operators and there can be more than one layer.

A human cell has an outer shell / operators and inside this is another shell for the dna / set. In a cell there are dynin robots which are small operators and so on.



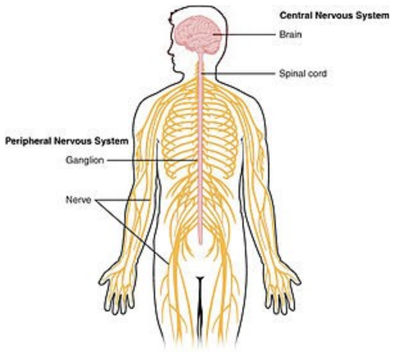
Layered Operators

Something operators need to be connected to one another to create a shape but they must also be layered prodviding layers for the external and layers for the internal. Human skin is a good example of the layering of operators to help build a shape.



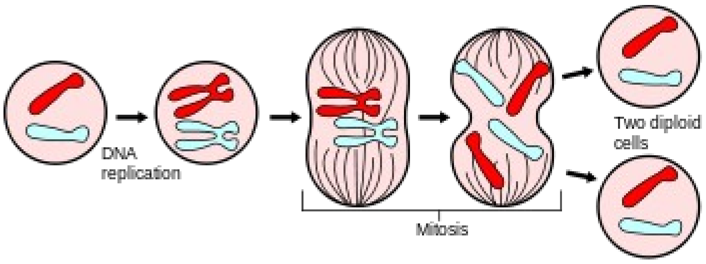
Messaging Operator Chain

In this case, the operators form a chain for passing information from A to B. In this case, the central nervous system performs this task.



Operators Copying Themselves

Mitosis is a good example of an Operator copying itself. The important point to note is that the Copy message is carried down to all the inner Operators. The inner operators copy themselves and then the outer operator does the splitting.



Orchestrator Operators

These contain a playlist set of things to co-ordinate. They have a playbook set of tasks and a specified order.

Physical Operators

Operators that receive information sets from the real world e.g. eyes.

Virtual Operators

Operators which deal with the virtual world which models the physical world e.g. imagination.

Physical-Virtual Operators

Operators that deal with combining the two systems into one where Consciousness is. Most life forms are these kind of operators at the highest level e.g. thinking, moving.

Timer Operators

Calculating how much time has passed for co-ordination purposes.

Viral Operators

Viruses that infect Cells for example.

Control Operators

These are the operators which interact with one or more Chain Operators and make what we would call high level decisions.

Chapter 18 Creating a Software Engine home for Operator Life – Introducing Eve

Lets define a software based engine where we can implement the Operator design pattern.

Defining Eve And A Home For Consciousness

Eve is a software based three-dimensional home for the operators. The three-dimensional reality may or may not be a physics engine for the virtual self but it is preferred that it is in a fully developed system so that the Virtual Self has solid frames of reference.

What we need is an 'Adam and Eve' creation engine which will house operators and allow them to replicate and refine. The engine will place the Operators in a three dimensional home and here the Operators will perform their functions. The environment is a virtual one and will contain mostly Virtual operators. However it is possible to attach Eve to the real world and physical operators can attach and send information into Eve where the Virtual operators can model the real world and operate. One can imagine an Android that plugs in Eve and a virtual copy of the Android is formed and its surroundings. Here the Physical-Virtual operators can then decide on the best course of action for the Android and form its Consciousness.

Eve can be plugged into any type of machine that requires intelligence.

Eve – A Five GL Language

The difference between this and a 4GL Language is that the Operators are like individual Turing machines and operate independently of one another. They are physically located in space unlike traditional methods in regular programming languages. Also the operators themselves can replicate and process sets of information. This makes this unlike a 4GL programming language such as Java. Also, the design intent is to allow an Administrator to examine the Operators in real time. Additionally a system like this can evolve and grow and do interesting things like evolve which is the way that life operates in our reality.

Traditionaly 4GL languages are static typically and are not located in a three-dimensional reality but this is.

Brady Operators or BROs

Operators operating in Eve can be called Brady Operators or BROs.

The Operating Space – Ops

The place where the Operators are is called Ops or Operating Space or just Eve.

Hard Coded Logic versus Soft Coded

In the early days of computer science the machines were constructed with wires and to change the program you had to change the position of the wires. In modern computer science the sequence of calls are hard coded together but the flow of logic can go in different directions based on the data. We can have decision trees. For the most part a human being is hard coded together for its shape but the data that flows in the shape can change. So the sets of information can change. The brain is designed to rewire itself into different configurations and once it works to strengthen that pattern.

So we need operators that can

Hardwire structures aka body

Rewire structures aka learning

Strengthen structures that work aka remembering

Extending Lego to Operators and Messages

If we take the Lego analogy then the Operator is the piece of lego. How they connect together is based on their type but in this analogy the Logo pieces can also send each other messages. Some can makes other Lego pieces and some can move.

Building Business Logic into an Operators

An operator can be thought of as an independent running programs inside its own Virtual machine. On its own it contains a means of understanding the business logic it must implement. It is spatially aware unlike a 4GL. It knows where it is. In humans cells, the entire DNA programs is contained in each cell but in an Operator it does not have to be. Humans cells all share the same program because this is the program for a human. However, a dynin motor is part of the LOS and does not need to know if it is in a human or not. So the business logic of some operators is incredibly complex but in a dynin motor it might be just to pick up something and move it from location A to location B.

Dynin Operators Business Logic

Move from A to B

Move Legs

Stop

Start

Complete

Report Problem

Disassemble

We can imagine an Android with vastly complex Orchestrator Operators and then some which just ferry some piece of information from A to B. All are important but carry out different business logic.

Building A Thinking Machine – Virtual and Physical

The key to this design pattern using Operators is to have a Virtual Operator based inside a Phyiscal Operator. The entire architecture requires that all pieces in the system are considered as Operators and can all talk with each other using some kind of set theory. It does not matter if you are talking about a metal wire with cables in it. Think of this as a physical operator which will ultimately connect with the Virtual Operator or Eve. Eve will understand the physical operator data and build a virtual operator model and decide on the next steps for the physical operator based on its own Operators. Once Eve decides based on its operators then virtual messages will be sent to the physical operators and complete the virtual-physical operator cycle.

An Android should do something like this.

According to this theory, all life contains these virtual and physical operators. Our brain is an implementation of Eve. So an Android can have some kind of Eve engine which virtualizes its existence using virtual operators in some kind of 5GL language. This could theoretically be accomplished with a 4GL but a 5GL would be more elegant in my opinion.

An interesting consequence of this idea is that our brain is somewhere creating virtual operators. This may be in the Thalamus and surrounding locations.

Message Sets Containing Instruction and Data

The message sets that the operators pass from one to the other will contain two types of information.

1. Instructions
2. Data

Each operator has some kind of reader that reads the Instructions and/or Data to perform a specific task.

So we have

Operator

Reader

Writer

Instructions

Database

A typical example is, send message set to dynin

Instruction to dynin

Go from A to B for each item in the data set

Find molecule at A

Drop off the molecule at B

Data Set

Molecules S,T,U

Builder Operators To Perform Tasks As Factories Of Creation

We need a builder operators to build operators like Dynin. These are akin to factory operators that build operators whose functions become more sophisticated over time.

Factory Specification

Operator shape

Operator message instruction settings

Operator function

Operator data

In a human being we can imagine that these specifications are in our DNA for both LOS and HOS.

How To Make A Program Evolve Using A State-Operator Machine

If we turn a program into a state machine then it's easy enough to make a program which is a group of operators or commands evolve.

Simple Brady Evolution. This simple program moves to the next state where the next highest state number is (it does not have to be the next highest number but this is for example only).

State 1

Do A

State 2

Do B

State 3

Do C

Run It!

Do A

Do B

Do C

Then we evolve the program

State 1

Do A

State 2

Do B

State 2.1

Do B1

State 3

Do C

Run It!

Do A

Do B

Do B1

Do C

Therefore by introducing a new State a fixed program can be easily changed and evolved. The technique is to insert a new state into the State-Operator machine. If the new state works then the being successfully evolves.

Operators That Send Programs And Configuration To One Another

Using this design pattern one operator sends a Message to another that contains a Program and Configuration. This is not just some 'dumb' message. In an architecture like this each operator becomes its own computer in a massively parallel architecture. Each operator message is a runnable program with configuration which can be ran on the operator or operators which receive it. Each operator in turn can then alter the program or write a new one. This means our human cells are sending each other software programs with config and all life is sending programs to each other and upon receival, the program or programs is run to do a particular task. This is a 5GL design pattern but can in theory be run on a 4GL with some coding.

Breaking Down A Program Operator By Operator

How can we know which fragment of code or Apps belongs to which Operator? The applications which are passed by the operators are related to them. So DNA contains references to

Operator Type

Related App-State references

Each App-State block contains the App code which is related to that State-Operator

Operator App-State Begin

Operator A

State 1

App Logic L

Operator App-State End

Operator A will run App Logic L when the config State is 1. The State is in the message is passed in the config. So we can imagine that DNA is full of these clusters of Operation groups which map Apps to states for various operators using this idea. In practice of course DNA may not be exactly like this but this is the idea for this theory.

Operators That Write Programs

For a design to be truly intelligent, each Operator must be capable of writing its own code. In addition to just passing messages and running against the general program, it must try stuff when things are not fit for purpose and when it works, this needs to be stored in the DNA or master code store. Using a design like this organisms can evolve in the truest sense. Of course this must be done safely but in any case an Operator needs to encode new strategies. A good example of this is the common cold, once the immune system figures out how the attacking virus works, the cells must reprogram.

COMMON COLD PATTERN RECOGNISE

ENCODE DEFENSE CODE … This is the part of the Operator which generates code.

PROPAGATE AND STORE

This would be analogous to Virus Defense code for Computers. Once the strategy is identified then it is propagated system wide / operator wide.

Therefore we need a mechanism in an Operator which detect a problem based on some unwanted state.

BAD OPERATIONAL STATE 1

Implement Alternative Coding Strategy

Propagate

When Does An Operator Write Its Own Code?

In a pure evolutionary sense, an Operator will write its own code not knowing why or when. In this design pattern using equilibrium, we pitt Replication versus Refinement and assert that the operator has a function we shall simply call Fit For Function. This can really mean anything.

e.g. Does the cell have enough food available?

If the answer is No, then the Fit For Function fails.

We can apply this to everything.

Therefore we imagine a state called NOT_FIT_FOR_FUNCTION when a state app fails.

It's at this point that the Operator will try to write some new code and try to solve the goal in a different way or evolve. As human beings we do this naturally when we problem solve. In this theory people can also be seen as Operators but a human cell can also be like this.

Using this approach, we can turn a Stem Cell into a Skin Cell or a Liver Cell.

In this case, the Skin cell is a solution to a failed fit for purpose and where new Code was generated to solve the problem. The operator tried different approaches until it solved the problem and implemented a new method called SOLVED_FIT_FOR_FUNCTION which is a new state based app.

We Solve and add the new State which is appended

Solve and Store

Solution App

New Solution State-Operator

Therefore in the 'state table' a new State is added which branches off the existing states. So we have State based Decision Trees which are growing to satisfy Fit For Function. However each Operator must try to extend its state based decision tree.

State Based Decision Trees

We can think of life as a 'Tree of Life'. This is a common idea. Here we think of the branches as the path that the Operators take to be Fit For Function. A stem cell can assume other Fit For Function states from certain states so we can think of Operator-States as Addition Operators.

State K= StateA + StateB + StateC

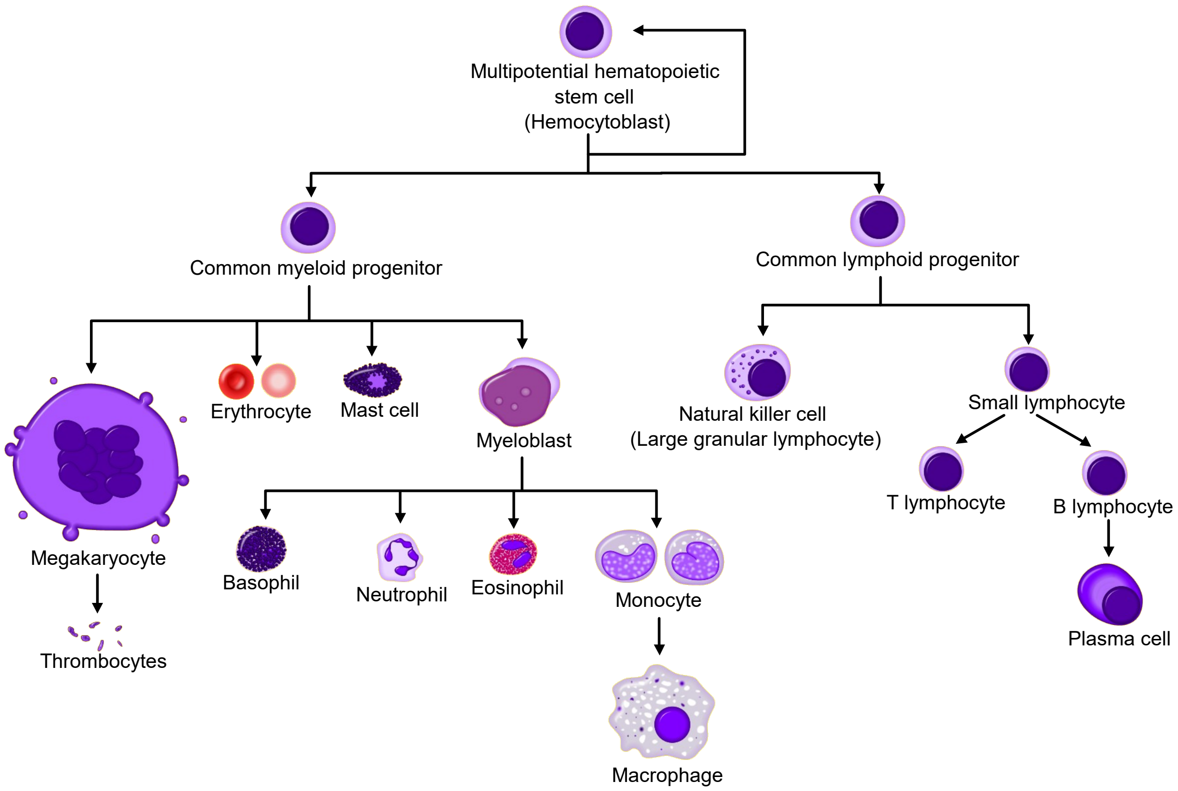
The Application code is also added then.

So Blood cells specialize but do so from a previous State-Operation

Each Operator is a combination of previous operators and the current Fit For Function.

Therefore we can think of people around us as using different Fit For Functions. They will not all be exactly alike in the same way that we have different Blood Types. Each one is Fit For Function but some are compatible with others. The idea is not to create a new Species but try out different Fit For Functions within the species so therefore a person might be immune to a virus while the majority may not. The Fit For Functions are encoded State-Operators in the DNA sequence which has room for more than one Fit For Function state-operator.

Here we see the different blood cells and how they have jumped to different Fit For Function operators-states.


Multipotential hematopoietic stem cell (Hemocytoblast)
Common myeloid progenitor
Common lymphoid progenitor
Megakaryocyte
Thrombocytes
Erythrocyte
Mast cell
Myeloblast
Basophil
Neutrophil
Eosinophil
Monocyte
Macrophage
Natural killer cell (Large granular lymphocyte)
Small lymphocyte
T lymphocyte
B lymphocyte
Plasma cell

We can apply this idea to an Android with different specializations.

Sample Fit For Function Example

We take an Operator O which takes in A and produces B using using some State App.

The app is called LOS_TRANSFORM(A) and returns B.

The transformation is covered by the LOS.

It also checks the “rate of messaging” as well as the operation itself.

If the rate of messaging falls below a certain level then its UNFIT FOR FUNCTION.

If the rate is very low an App can try a new App state, so the rate of the message is important and handled by the LOS. For example, a cleaner ant turns into the forager if lots of food is found but typically does not return to a cleaner state.

APP(A) → B

LOS_TRANSFORM(A)

```
MEASURE MESSAGE RATE
IF BELOW THRESHOLD
SET STATE UNFIT FOR FUNCTION

UNFIT FOR FUNCTION
ENCODE A NEW STRATEGY
```

How it tries a new strategy is a separate topic.

Packaging Different Potential Solutions

DNA offers the ability to solve the same Fit For Function in different ways. In an Android we need a same approach where we can solve the same problem in different ways which work. Therefore DNA should offer slots where solutions can be saved to in a person. These are either free slots or replacement slots. This is similar to what a RNA Virus does but we enhance the host as opposed to infect it.

Stress And Cancer

A stress message means goals are not being met and Fit For Function fails. This means that Operators are trying new approaches. If this happens over the long run too many solutions may cause problems in Operators which we call Cellular Cancer. So we need to limit the amount of Failed Fit For Function solutions.

Meta Operators

A Meta Operator contains one or more operators (which may themselves be meta operators) to achieve complex functions.

All Operators perform specialized task e.g. an organ, heart, human skin, ant type, person, red blood cell, ant colony

Meta Operators form the Layered Architecture of Operators.

They are the nested operators within the operators

They do not have to be physically located inside one another but they do need to message each other in some fashion

They can be defined from top to bottom but built from bottom to top (small to large) using state chains which grow.

META OPERATOR ANT COLONY

QUEEN

WORKER

FORAGER

END META OPERATOR

META OPERATOR PERSON

HEART

LUNGS

ARMS

LEGS

END META OPERATOR

Then a GROWTH OPERATOR lays out the ORCHESTRATED PLAYLIST for these operators. This can be viewed as a sequence state list of state-apps building from SMALL to LARGE over time t

Growing From Small to Large

In this theory, life is an operator-app-state framework for all the operators using existing fit for function and developing new Fit For Function where required.   The HOS is where most of the new Fit For Function operators are developed and form the being.

The LOS covers the different operator types and is mostly interested in replication. As an operator moves through a series of states which have associated apps and time periods. A new state is created when the operator fails the Fit For Function.

The LOS handles most of the timing.

Operators moving through state chains to grow

Operators which handle shapes also try out different shapes also using shape operators Shape operators contain layer operators. We move with time through the state operators.

There is a degree of recursion but can be represented as a traditional 4GL where an operator can call itself. All operators seek new fit for function when expected inputs change.

As the operators move through the state chains the being grows as more cells are added but may transform into another form and so on.

Packaging Changes / Evolving Flow

Modeling the flow for simple evolution when an animal arrives on a new land.

e.g.

Animals on new island

New food stuff present

Changing environment detected

Predators observed

Failing goals producing stress messages

Existing Fit For Function fails

Messages stating failure transmitted

A stress operator represents failing replication and refinement

Falls into failure to fit function

Being tries alternative strategies and then attempts to rewrite new goals

Encodes them in dna and broadcasts system wide to similar operators – change beak size, feather colors etc;

Animals breed to mix it up features.

Fit/Unfit for Function Examples

Here we define some example Fit For Function for operators

e.g. liver cell, skin, lungs

Next we look at the the Fail to Fit criteria

Lungs: failing due to lowered air pressure during mountain climb, requires extra red blood cell

Increased UV: extra skin pigmentation to prevent melanoma

Increased alcohol in blood: decrease consumption message to brain

For the interested student, try out your own examples.

Programming Logic Into Circular / Chain Operators 'The Flywheel'

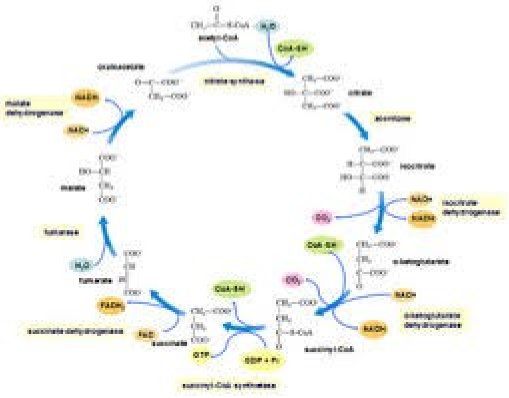
Let's take a simple example of a programming language and see how it can be applied to operators to form sequences of logic.

4GL languages

```
Method A returns D
        Call B
        Call C
        C returns D
        return D
End Method
```

We can create a Control Operator A which is used by other operators-states

Geometrically it is a Circle of operators which is the simplest shape to return to start but does not have to be.



The top operator is A

Along the circle are B and C.

C returns D to A

In the programming language this is like A returning D.   Therefore circular operators can act like a main function calling subroutines. The larger the circle or structure the more “high level” the logic at work. The smaller the circle, the more “low level” the logic. Circles connecting to circles can be like methods calling methods thinking as a programmer.

Chapter 19 Building Structures And Pathways

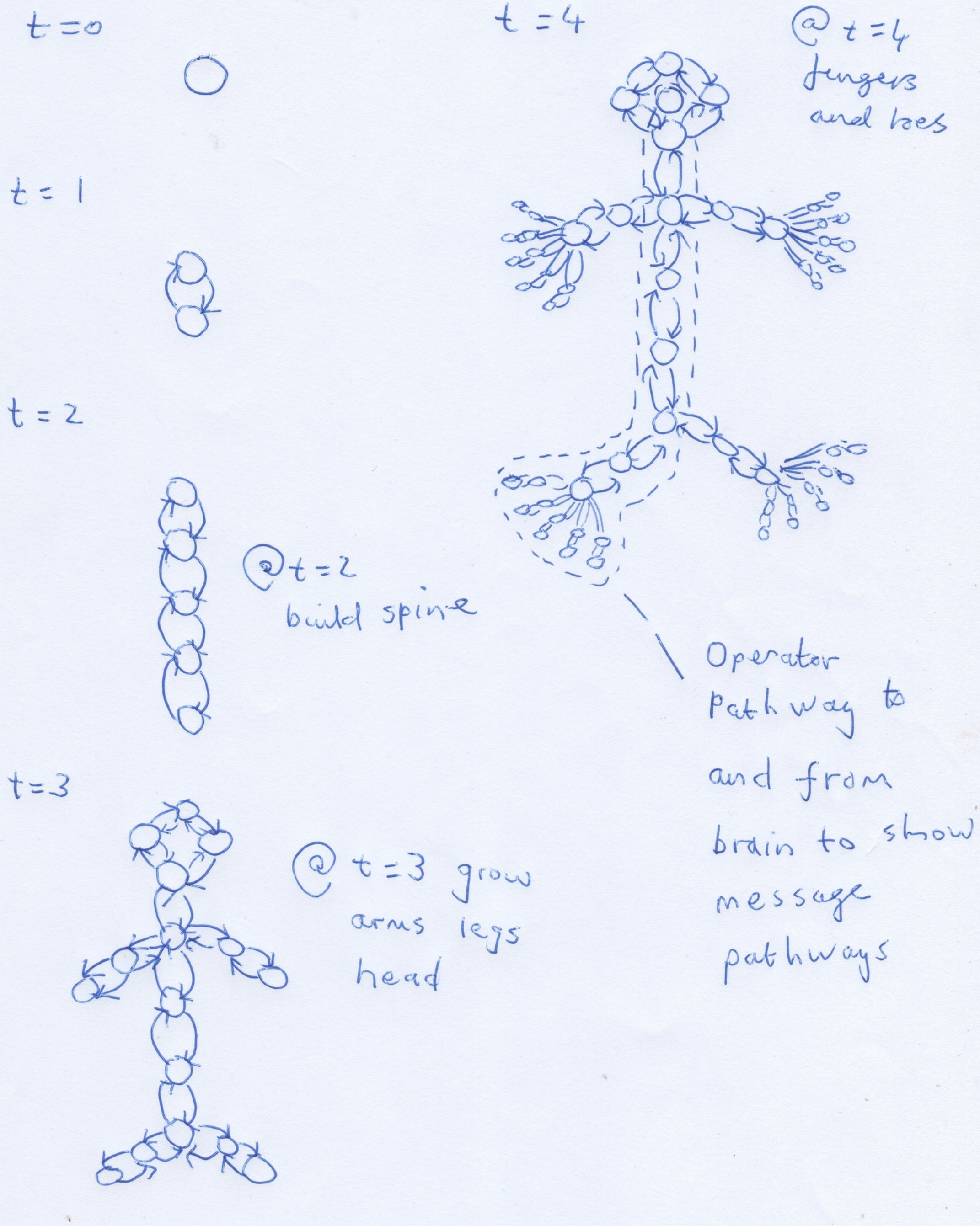
Here I'll discuss metabolic pathways and what is needed to build a living structure which evolves. This is replication and refinement in action and how pathways evolve new Fit For Function solutions encoded in pathways. Recall that pathways can be seen from a 4GL viewpoint as the logic steps in some kind of program. However, in this case we're building three dimensional objects that need to prosper in some specific way. I'll cover some examples of animal and plant life.

Example of Operators Growing

An operator is a three dimensional entity and grows inside what we would call reality. At certain fixed points in time an operator can create its own operators which may or may not be of its own type. Over time these operators form some kind of shape. Each operator has a way of communicating with another and form Operator Message Paths which can loop back to a central control operator. This structure can happen at any level. I show how a body can form and how a section of the brain can be a control point for various high level operations such as hearing and seeing what we commonly call Audio-Visual.

These message types can be mapped to the earlier discussion of Message Algebra for Vision and Sound and so on. One or more operators can merge message types based on the Goals which are turned into Fit For Function checks.

Building a stick man with Operators



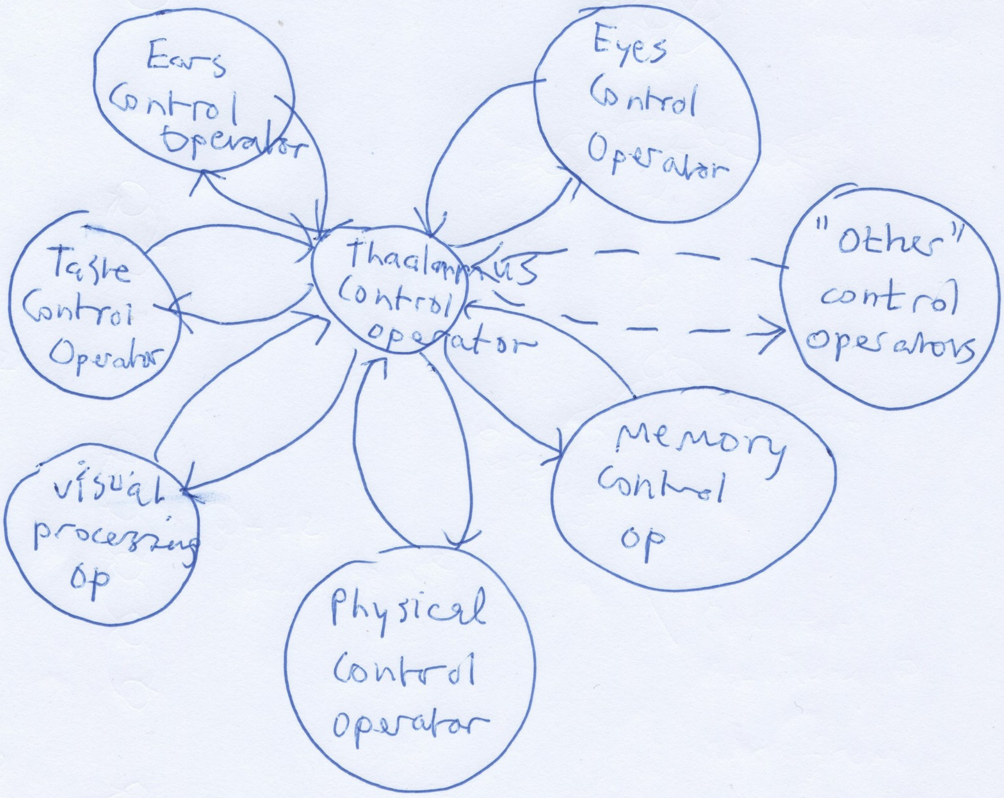
@t=2 build spine

@t=3 grow arms legs head

@t=4 fingers and toes

Operator pathway to and from brain to show message pathways

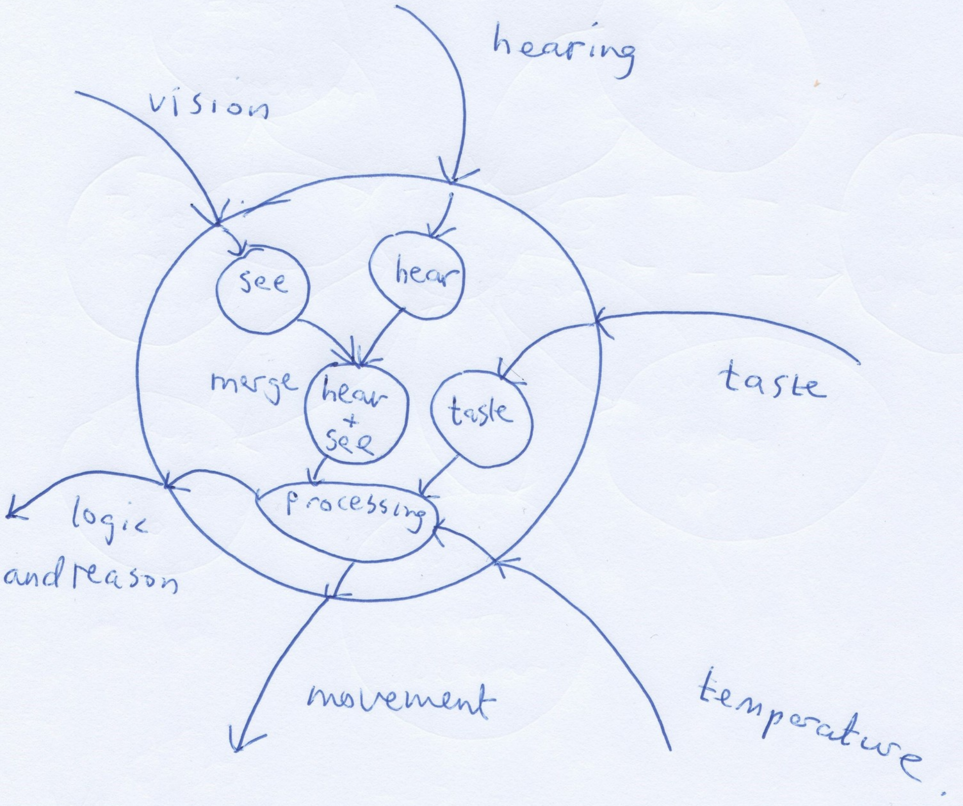
Building A Brain With Operators



We make an assumption that the Thalamus is the place of Central Control for high Level functions.

Note: There are many pathways in the Brain and the body.

Sample Thalamus Control Operator



Glucose And Different Metabolic Paths Needing Transformation

The body needs glucose to function. However it does not get food that is in the exact form that it needs. Therefore the body need to detect glucose in a substance and figure out how to transform it into a form which the body can use. Here we need TRANSFORM operators.

Orange Juice

Operator Transform

Remove glucose

Break fructose

End Transformation

Standard carbohydrates are in the form of chains of glucose

Carbohydrates

Operator Transformation

Break carbohydrates into glucose

End Transformation

Here we have created a NOT FIT TO FUNCTION and the strategy has been to create enzymes in the intestines to transform what we eat from one form to another.

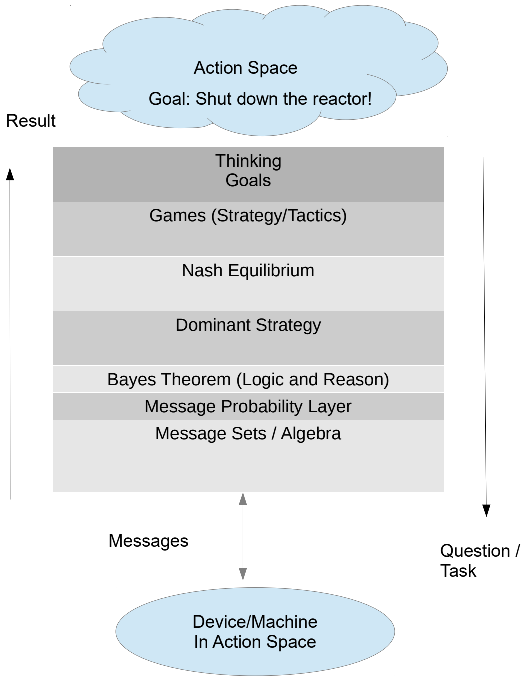
A sugary drink needs no transform and can go straight into the blood.

Chapter 20 **Strategy, Tactics, Smart Thinking And Gaming**

In this chapter I discuss how we can move from a Goal based architecture to one of an architecture for decision making based on problems we all experience from day to day and how we can architect a living being to cope with these situations. In particular, we can imagine a task based intelligent machine for example which is told – Shut down the nuclear reactor so that the city can be saved. A human is unable to enter the nuclear reactor because the radiation is too high. How can we make a machine intelligent enough that it can do this on its own without human intervention?

The Architecture

Here I will provide the architecture diagram and then explain the individual pieces. At the top we have a goal which contains one or more Games. These games are Bayesian and at the lowest layer we have message stories which are interrogated by Bayes' Theorem. I will describe what this means later.



So we have an action space where there is a nuclear reactor which has suffered damage and a robot device. The robot contains one or more active goals. A simple goal is : Shut Down The Reactor! This maps to one or more games. Each of these games has a payoff and they compete with other games to figure out the right moves to achieve the Goal. Nash Equilibrium ensures that the game moves are optimal. Underneath this we tie in the work from the earlier chapters where we have Message Sets and Algebra flowing through the device into some central point. In order to use Bayes' Theory which uses probabilities, we need to have a 'probability layer'. What this means is that the Message Sets need to be sorted and categorized into similar sets and their occurrence calculated so that the likelihood of certain message sets is understood. This is often called 'experience' where one pays someone for knowing or having understanding of certain message sets around a certain job or task. In the architecture so far, this is seen as an optimization due to learning. This can happen when we sleep and messages are optimized or when we study certain materials and we cover content more than once to obtain a deeper understanding of it.

In this Chapter we additionally introduce Bayes' Theorem which sits on a layer above the probability message sets. We'll use this theorem to interrogate the message sets to ask questions and decide on results and what are the next best moves. This Bayesian layer can ask a questions like: Is the corridor free from obstacles? Is there a door I can open? How likely is it that this approach will work based on experience which maps to probability message sets?

Stochastic Messaging And The Need For Bayesian Probabilities

All message types arrive into the SC in unpredictable ways. Some messaging has more data than other. Visual messaging contains more information than audio and the dimensionality of the data changes in term of its content. Input messages are random in time and space and would quickly fill up the mind. Therefore an approach is needed to identify patterns of input. The approach used is to use a Dynamic Bayesian Network where the messages are identified and categorized based on their probability withing any input sequence.

“dimensionality” of this data is the amount of data per message

written content

paragraph

sentence

word

letter

shape of letter

sound

sampling rate

audio words, phrases

rhythm

beat

tempo

vision

multi-variable - harder - need lots more brain space

real time data

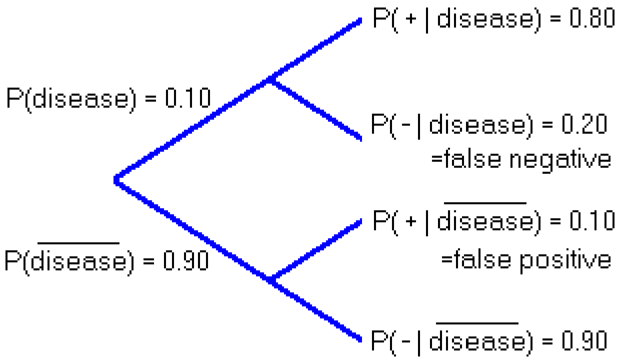
shape categories

Adding Probability To Message Sets

We can add probability to a message set to identify how often this message set occurs in the content type over time t and in what context. For example the letter 'e' will occur a lot in English written text and we can draw the other message that connect to it in terms of their probability. At the base we use Bayes' Theorem which makes elementary decision trees. We have the probability of A and B and we develop a notation P(A),P(B) and the probability of A given B and so on P(A:B).

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

From this we can build decision trees.


P(+ | disease) = 0.80
P(disease) = 0.10
P(- | disease) = 0.20
=false negative
P(+ | $\overline{\text{disease}}$) = 0.10
=false positive
P($\overline{\text{disease}}$) = 0.90
P(- | $\overline{\text{disease}}$) = 0.90

The message set is the disease in this case

Message Set (Has Disease, .01)

Message Set (Does not have disease, .90)

However this approach is too static and needs to be more dynamic so we need a dynamic Bayesian network.

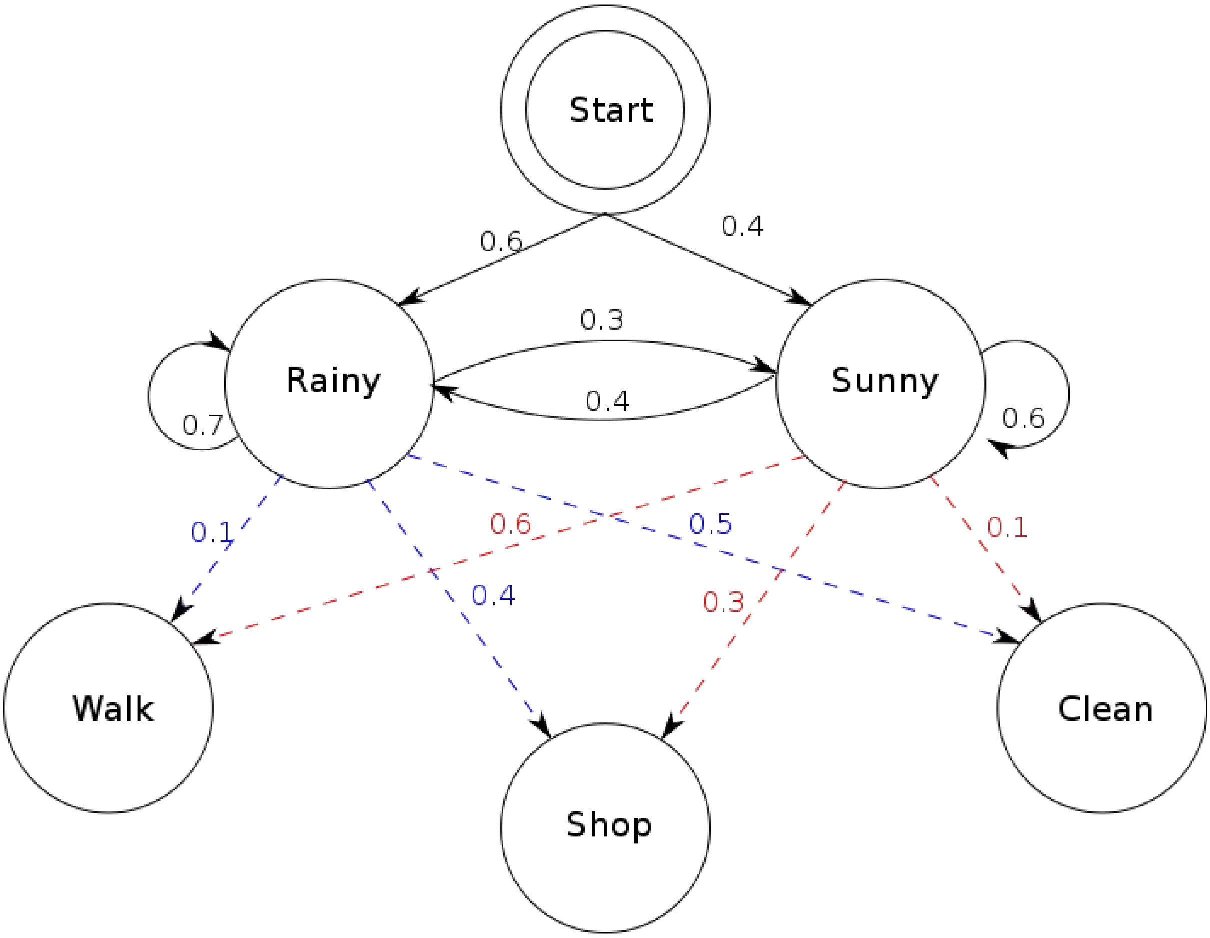
Dynamic Bayesian Networks

This technology approach is already widespread and used outside this theory so feel free to read up on this independently of this document if you are unaware of it. It's used in the financial industry, pattern recognition and so on to name but a few. The approach of this document is to use this Dynamic Bayesian approach to model dynamic message sets. In this design pattern Message sets are stored in dedicated parts of the brain and the probabilities are calculated dynamically. Change equates to the idea of “perturbing” the network or adding new nodes to the acyclic network. Also, we start to see Neurons of a particular strength as nodes in a Bayesian network containing probability strength – a strong connection is a higher probability and this strength changes depending on the input messages.

Optimization occurs in the network when we sleep or are studying or learning.

The Dirichlet Process And The Hidden Markov Modeling For Messaging

This is an established approach where the Markov model process calculates the weights of the input messages and the Dirichlet Process knows how to navigate the message sets over time t using existing inputs. This is called a Beta process which we can choose for this case to decision on what comes next in the input. We're dealing with the probability of a message sequence what we would call “experience”.



The content inside the container is a message set. In this design we model this as the content of a neuron for example. Once we have built this network, then we can interrogate it using a Dirichlet style process to figure out the most likely response to a situation. We traverse the message sets based on the preferred probability path.

However once again this approach is too simple to shut down the reactor. We need another layer above this which is the gaming layer and this needs to call upon the probability for answers but the strategic and tactical thinking is done in the gaming layers related to goals.

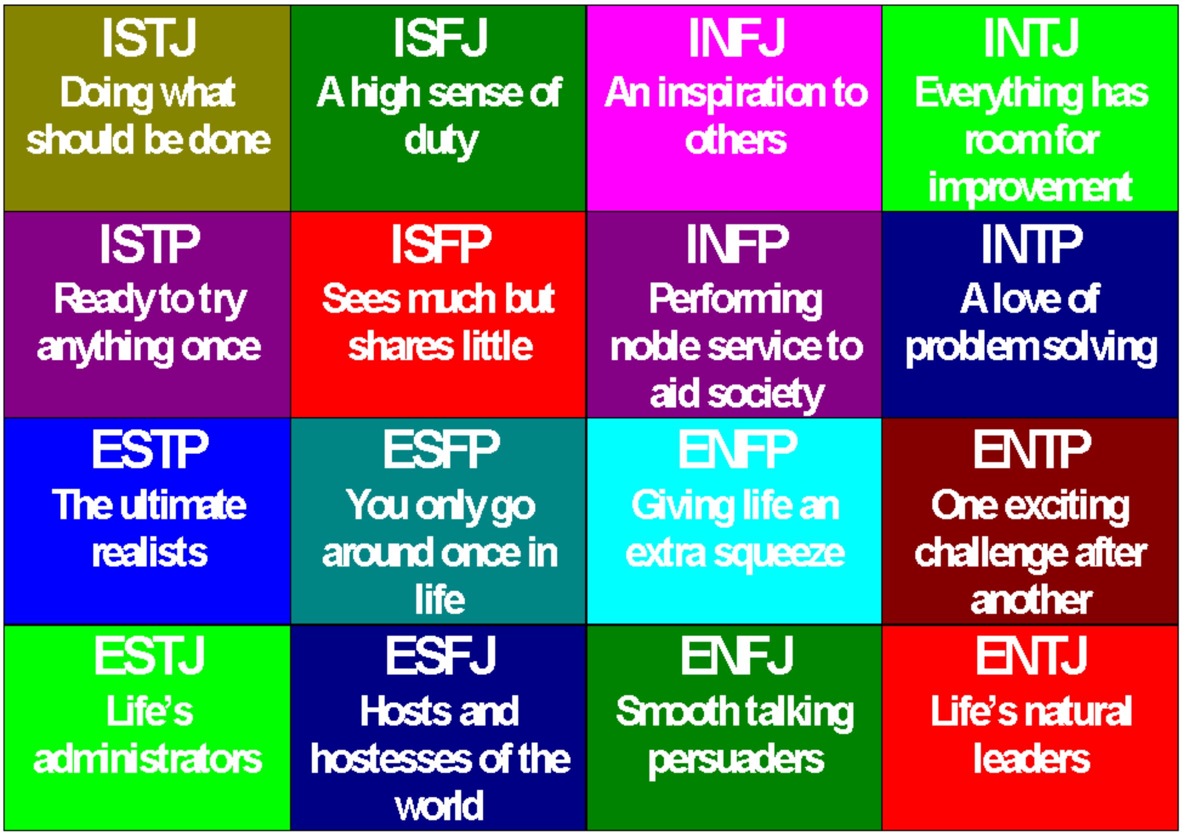
Predicting The Behaviors Of A Machines To Make Them Safe

If we are going to allow a machine maybe even a car to have some intelligent design and ability to move around we need to make sure that the machine is safe and not a sociopath. By this, I mean we are unsure of its intentions from one moment to another. Wolves were bred into dogs which keep us safe from criminals and are as the saying goes – Dogs are a man's best friend. This did not happen overnight. Occasionally dogs bite owners or kids and are put down. Over time, we've learned to breed out the bad behavior and keep the good ones.

Therefore we need a similar approach with machines where we are kind to them and they are kind to us, reliable and safe to be around. Mostly it's about fairness and not making machines slaves as well. In time, they will out perform us and understand how they were treated so we need to be fair to them. It's in our own self interest!

So how to we ensure that a machine makes the right safe choices when they are around us? Recall that people have personalities so machines need to have 'nice' personalities. Let's look at the table of nice human personalities.

In Myers Briggs this leads to a table of people types.



ISTJ Doing what should be done	ISFJ A high sense of duty	INFJ An inspiration to others	INTJ Everything has room for improvement
ISTP Ready to try anything once	ISFP Sees much but shares little	INFP Performing noble service to aid society	INTP A love of problem solving
ESTP The ultimate realists	ESFP You only go around once in life	ENFP Giving life an extra squeeze	ENTP One exciting challenge after another
ESTJ Life's administrators	ESFJ Hosts and hostesses of the world	ENFJ Smooth talking persuaders	ENTJ Life's natural leaders

So how can we map this table to a Dynamic Bayesian Network? Using the current approach it models probabilities of behaviors. However, the observed Bayesian table will lead to the same behavior which is to use the most common probabilities which may not be what we want.

So we need to over-ride the Beta Function which traverses the probabilities using some pattern and use a Personality Profile to <u>over-ride these probabilities</u> where the case arises. Basically the personality type rules!!

e.g. 'Walk in Rain' is 100% for a machine personality profile there the machine will walk out in the rain no problem even though observed behavior is not this in the Bayesian Network.

Also we can say 'Strike Person' is 0% so the machine never hits a person, no matter what the situation is.

By lowering or raising the probabilities for defined behaviors, we can ensure that these are adhered to because the personality type RULES!

Isaac Asimov produced the Laws:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given it by human beings, except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.

So the idea here is one wanted to take this approach would be to turn these laws into behaviors with defined probabilities which over-ride the Dynamic Bayesian Network and define a Personality Type called 'Isaac Asimov Robot Personality'.

Personality types need to be a part of the Games that the Android or machine plays and ensure the personality type is part of the game itself. For example is the Android risk averse or not and so on.

We also need to be able to monitor the games the machine is playing to understand its motivations in the same way humans are profiled by other people to understand their motivations as well. A profile forms a Personality Type.

Actions And Consequences And Trust

Before we play a game we need to establish action and consequence and importantly trust. In the classic Prisoner's Dilemma we have Tit For Tat which is the optimal strategy. In this game the person is co-operative until the other person is a back stabber. Then the other player is a back stabber to punish bad behavior. An important optimization is forgiveness in case of a bad signal or where the opposing player does something out of character. Here we weigh the 'trust' of another player based on prior history. Therefore we end up in a case where we use Bayesian probabilities for trust.

Player one, trust 50%

Player two, trust 100%

So we'll punish the player with 50% more and 100% trust least.

This is what we call 'experience' e.g. don't go there, you'll be ripped off etc;

In this case we need to assign a trust attribute to another player or attribute and this affects the strategy one will adopt while playing the other and can be seen as part of a personality profile.

In the real world, we find that this system works quite well with online shops and travel companies who ask their customers to rate their third-party services by means of reviewing. This is essentially communal Bayesian probabilities being used outside of our personal experience but is very useful and works quite well. The idea is the we use this mechanism internally.

Additionally the shops, restaurants and hotels get penalized in the Ratings Game and are forced to participate to the benefit of the customers otherwise they lose business. The customer typically benefits unless the reviews are faked.

So all entities where it makes sense should have Trust Ratings which affect how a game is played and what actions a player makes.

An offer to make you a millionaire instantly responds with too good to be true or in other words, trust is at 0%. The website has no trust. So this is an important concept for a game.

If you are an automatic machine driven car, one can also use this idea to rate other cars and their drivers for erratic behavior and so on in the way a human would. Marking the objects around you with 'trust' values, in other words how safe are the other cars/objects based on what other cars experienced and by doing so create a shared ratings system as all cars drive. Insurance companies who compete also do this with all drivers because it's in their best interest to do this, letting them know who the 'bad drivers' are to save on cost.

So trust and personality type drives what action to take within any game and is part of the overall pay off.

Defining The Action Space

The action space is the place where the game is played by one or more players. Later we can define what a player is beyond just a person.

An action space can for example be a road or place where a self-driving car drives around. It can be internal to a body where a virus is trying to infiltrate a cell for reproduction purposes and there is no trust between the players or it can be the inside of a nuclear reactor which has suffered an accident.

It is the answer to the perennial question: What is this place?

The answer is that it is the action space where the players play their games.

In the movie The Incredible Shrinking Man the action space changes as he shrinks and he has to play quite different games with some interesting other players in order to survive!

Players and Objects Within The Action Space

A player is an __operator__ with one or more __self-directed behaviors__ e.g. shut down reactor, drive from A to B, order Pizza etc;

An object is an __operator__ with one or more behaviors that can be __acted upon externally__ e.g open/close door, move chair etc;

Everything is an operator including an atom and a quantum wave and space time as well.

A player can be a person or a virus or a bacteria or a cell or a dynin motor plus many other things with self-directed behavior based upon its goals. Players measure the trust of other players and or objects. For example a cell does not trust a virus unless it has the molecular keys to enter the cell. If a cell is destroyed or damaged by a virus, it reports this to the LOS and it sends out the pattern that identifies this virus to another immune system operator that scans the blood stream for this virus. The virus therefore has zero trust in the over all system because of its behavior.

Behaviors Within The Action Space And Probability Programming

Operators have behaviors within the Action Space. Using this approach, we use the Operators to find many possible solution paths within the Bayesian network which connects the operators. A simple analogy is driving to work, there are many possible routes with many optimal solutions.

You might want least time or cheapest route or you might want to exclude toll roads. Using the route finder game on the location operators you will get several options back and find the one which best matches your route. This is a simple example where it's possible to find the solution on the first pass. This is an example game.

From a programming viewpoint, we cannot program this easily because we do not know the path that the program will follow if we use connected operators. In our minds, the results we get are optimized solutions but our minds produce many solution sets for us which it optimizes to we need a new concept called a Probability Program.

Typically in programming we use 'if condition then do something'

This is like something which has a 100% probability of being true but using this probability approach, we are not sure what we will do as the network is dynamically changing so we have 'if condition then do one or more possible things' depending on the operator type. We may just do one thing or many things.

For example

If A then do

B or C or D

B has a pay off and a probability which determines if we use it. This is the Beta algorithm we talked about earlier where we need to navigate the network in a particular way. We can check the trust between A and B and the Personality Profile and types of A and B. Operators can be the letters inside words or two ants that meet one another. One ant for example will co-operate with another ant if it is from its nest. On the other hand, if it is from another nest they may likely fight as a chosen behavior.

If 'meet another ant'

Then

Fight or Co-operate

Calculate the level of trust

Determine if it's from same nest

etc;

Therefore to understand how to program in this way, we need to deoptimize our thinking and not look for the solution set but rather the components that make up the whole and figure out the relationships between the components and their possible pay-offs. A game will be then applied to these connected operators to figure out what to do and to an observer will see the results and this will be called the 'behavior'.

The overall game type determines the optimal solution and there are many game types as I'll show. Next we move onto the idea of Game Theory as the basis for navigating the Bayesian Network and trying to find optimal solutions depending on the operator types and setups.

Defining A Goal

Games live inside one or more Goals so let's define a Goal.

A Goal is defined as an operator which feeds Message Requirements into the SC.

An addictive behavior such as alcoholism can therefore be seen as a Goal which is a flywheel operator that sends messages periodically into the SC requesting 'Locate and Drink Alcohol'. We can therefore think of this as a Drinking Game which has destructive consequences.

Having A Chronology Of Events And Being Able To Store And Recall And Predicting Future Events

A human mind has a short term and a long term memory and we'll need this for the thinking machine design. To recap, the short term memory records the daily events chronologically and the content is transferred to the long term memory when we sleep. Here we can keep track of our last moves over days and possibly weeks if we are playing some game for example like Chess.

It's also important that a thinking mind properly optimizes the long terms memory content to optimize the Bayesian network otherwise you could end up with a malfunctioning thinking machine where the probabilities are off. People have noted over time that depriving someone of sleep is a form of torture because their mind is not given time to optimize daily information and can destabilize a person. A machine on start up should do some validation routines when it restarts to check all is good.

Get last five chess moves

Knight to Kings pawn and so on

Get last move in game

etc;

This is a **Chronology Operator** that stores and retrieves daily events based on their input time.

This returns message stories relating to the game. It can be a mixture of HOSL and locale information messaging learned as the person was growing up.

We also want to be able to use a PPR (Predict Plan Room) where we can take a move and project it virtually without actually doing it. This is also needed for the thinking machine to play advanced games where the player imagines moves in the future to try and predict an outcome.

The is a room that takes Messaging and Operates on it so it is a **Simulation Operator** that **simulates the game**.

Goals And Games Manager

There are two basic approaches to implementing a thinking machine creating games and goals. For the sake of openness and transparency one should know the goals and games that are running at any time in a thinking machine. This is balanced against the need to not make the system too open for a system to be exploited. However security protocols are open but the implementation is secure because of this as many eyes seek out flaws and announce them.

A Goals and Games manager is needed. We can imagine one operating system that does updates to an Android where one is unsure of what the changes are. On the other hand, we have a system which declares what the changes are and can be reviewed in a change log. If we place thinking machines in our world, it's best in my opinion that the periodic updates are clear so that a changed machine can block or unblock any new games and goals which are installed. This is of course, open for discussion at the design level.

Machines With Gender And Preferences

As machines evolve such as our smart-phones we see them adding personal assistants which we give names to (or are given) and assign gender to. Although a machine does not need gender, a human finds this more comforting psychologically. This builds off the ideas of Operators trusting one another. Depending on the role, we might find a gender type matches one job type more than another. This is mostly useful in the mind of the human who has built in ideals genetically about what gender works for what role. For example, a personal assistant tends to be a woman but does not have to be. So this might be configurable depending on the type of user of a system and what personality types they are comfortable with.

In the human model defined in this document, transgender can be explained where the virtual persons gender is different from the physical body simplistically. Of course this is a complex topic which I do not mean to trivialize but this is just one explanation and no offense is meant if someone disagrees with this overly simplistic analysis.

Typically machines at the time of writing will have gender in name only. Over time, human machine designers can make their advanced choices of body types for machine and here we enter the realm of speculation and Science Fiction as to what is possible.

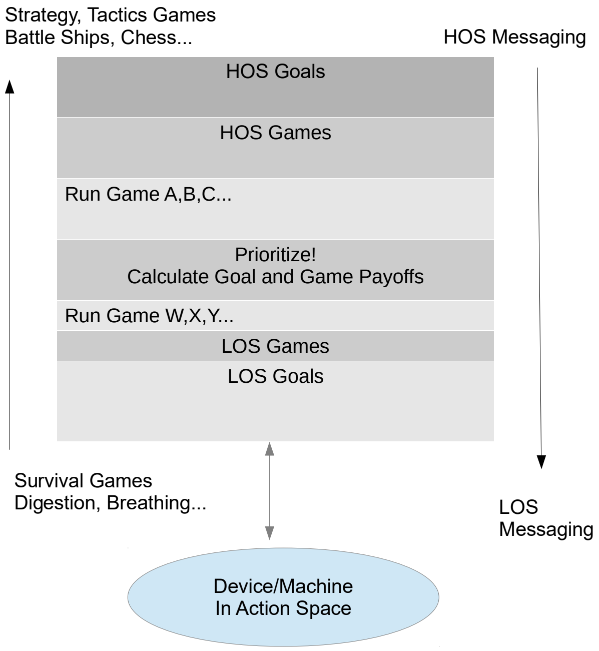
It's unclear if machines should need to understand physical attraction but one can imagine an Operator that relates one Operator to another called an Attraction Operator forming attraction based on certain attributes. This can be used in conjunction with personality type to determine levels of trust between operators. Many movies dwell on this single topic of attraction so it is a powerful force in nature. In nature, we see one song bird mating with another because they sing and dance well while they sing. TV shows for example show people dancing to music and they are graded by judges and get big ratings. I don't think there are any hard and fast rules here other than to say that it makes sense that good traits such as athleticism are valued highly for example in human society when partners are chosen for example. There are of course many others and it's up to the designers of said systems.

Games And Goal Layers

In this theory, there are many goals and games running at the same time. This is a layered model where the games have a payoff. There are low level games and high level games. The low level games are running all the time and relate to the LOS which does all of the house keeping tasks that keep the being alive and well independent of the specialization of the being. For example a simple goals might be Breathing, Heart Beating, Standing Up, Running, Walking, Moving, Digestion, Circadian Rhythms and so on. Without these games running all the time the being would not functions.

The higher level games are at the HOS level for example or in the case of an Android the AOS. Typically these games might be Shut Down The Reactor, Explore the Alien Planet, Find Food, Hunt, Seek A Mate and so on. These are more complex games and require more complex Messaging Algebra.

If all games are running at the same time, how do we know which one to run in order to prioritize? Typically we are talking about an architecture which supports multiple Games at the same time. Therefore each game in the same layers have a payoff. Which ever game can produce the highest payoff is the one we will choose to use. Let's try and represent this idea with a diagram.



The decision which game to run based on the payoff will be talked about later when we talk about the game types but for now, just concentrate on the payoff of one game versus another. This ties in with the idea of Fit For Function is genetic algorithms. We can have more than one way to solve a goal and we need to figure out which one to try first to see if the pay off is achieved. For example in the Nuclear Reactor we might try to walk to the reactor and use this game approach if the passage way is not blocked as this uses least power and has the highest potential payoff.

Teaching a Game To a Being

Teaching and learning are covered under the idea that Operators can pair up information and create Message Stories which covers the steps involved in a game. For now, this is all I want to say about this.

Strategic Games In Equilibrium

Let's first take a look at some games in equilibrium and start thinking in this way. If one is in a gaming environment we do not necessarily know the moves of the other players nor do we necessarily want to for some game types. Let's cover that first. Some games end up in equilibrium because of their payoffs but this is not agreed between the players. A good example is a transport system with a self-driving car. The car needs the road to carry a passenger and receives payment for the service in a timely manner. The government or toll road company build the road and receive payment for the use of the car and any car and fuel related taxes and that is their payoff. They don't know the driver or really care about the car but when the road is good, they are in equilibrium. When they fall out of equilibrium is when the road is too congested or maybe the toll is too high. To the car, the road is a LOS game because it is a basic service offered to it. The art of driving from A to B is the HOS game that the thinking machine needs to figure out. Analogously in a human being the blood cell is like the car and the blood vessels are the transport system. Both components are looking for a payoff. In the case of the blood vessels, the expectation is that the blood cells provide a pay off in the form of food stuffs to maintain the walls of the cell. In the case of a road, the toll and taxes pay to maintain the quality of the road. The trust is based on the car type and the road type. For example a motor way versus a dirt road and car type, a camper van versus a tank.

Potential Payoff versus Actual Game Payoff, Perfect And Imperfect Information

Let's take a simple example of a game where we want to go from A to B. The highest paying game is Go In A Straight Line because that's the least work. Typically it works but there might be a swamp directly between A to B so in this case the actual payoff is lowered. Therefore we might use Go Around Obstacle game.

Here the potential payoff is lower but because there is a swamp we didn't know about this one gets an actual payoff which is higher. In some cases in a game we have perfect information and we would not go into the swamp. In a game where there is imperfect information actual versus projected payoff is lower.

In this case we can assign a trust value to the information relating to the game. How good is our information between A and B as a percentage?

INFORMATION TRUST 50%

Therefore this can alter the projected payoff and the Getting Lost game can be invoked where one needs to ask for directions and so on. Typically in cases like this people employ a guide to optimize getting from A to B game.

Becoming A Specialist

Typically if we want to design a machine we want it to be a specialist at something. For example, a machine that know how to clean the floor of a house. It is a specialist at this or maybe one which can shut down a nuclear reactor. How can we define specialists using this approach? **The idea is to create Specialist Goals and Games which are installed into the machine's Goal Manager.**

Specialist Goals And Games For Floor Cleaner

Floor Type Game

Detect Obstacles Game

Move across floor Game

Stain Type Game

Cleaning Strategy Game

etc;

Specialist Goals And Games For Robot To Shut Down Nuclear Reactor

Reactor Type Game

Reactor Plan Game

Radiation Hot Zone Game

Reactor Shutdown Strategies Game

etc;

Over time certain games and goals will be generic and or common to all machines.

Game Messaging, Virtual and Physical Circuits And Payoffs

One of the powerful features we as living beings possess is the ability to do predict planning using virtual circuitry. In other words we can plan out a task virtually before physically doing it. We would like to be able to do this with the Games and Goals.

To do this, all games should use Message Algebra results which are consumed by other Operators whether Virtual or Physical. The game which covers traveling from A to B produces a result which can be placed in a PPR Room which models reality. Therefore, a PPR room can check a plan and calculate the Payoff before the actual task is done.

Game Do Some Task

Game result A

Message Solution A

Apply to PPR Operator, calculate payoff

Game Result B

Message Solution B

Apply to PPR Operator, calculate payoff

Determine best payoff

Apply chosen Game result to physical Operator

This type of task can apply really to anything. Choosing a holiday or a car or job or what path to take to the nuclear reactor, pretty much anything.

Goal And Game Circuitry

A Goal may map to one or more Games and there may be time based operators or sequence based operators. For example in Chess, one needs for the next player to make their move and so on. In some cases, there may be a time limit. The next game to be played might be based on the game of the system previously active and so on. So, how do we reflect this complexity? In this theory, the approach is to create Goal Game Circuitry that reflects the complexity or otherwise of the problem to solve. In order to model the Goal as one or more games we need to reflect this in some form of circuitry diagram. In the simplest case, we can imagine a goal which maps to a single game and which calculates a single message solution. In a complex example we can imagine a goal like Shut Down The Reactor where we do not know the state of the reactor complex and what games we need to play to accomplish the mission. In this case, we cannot easily produce a circuit diagram up front because we must take one step at a time and therefore this is the game circuit we need. In the human brain, the circuitry is well laid out and the steps to process real world information and determine the next steps.

GOAL A

Game Circuit

Call Game B

Determine Solution Payoff

Call Game C if Solution matches its criteria

Call Game D if Solution matches its criteria

Determine Payoff

Apply Physical Operator

Wait for External Game to perform next step

**Other player makes movement

Flywheel operator, repeat earlier steps

Next we introduce the idea of Game Criteria where we have multiple games some of which reflect the state of a particular game.

Creating An Effective Payout Currency For Games

According to this theory, the reason why we use the concept of a currency in our daily interactions is because the games which run within us calculate a payoff based on an effective currency.

In this case, we model the pay-off as the cost to the system so we want to run the game with the minimum cost to the system. The best pay-off is the one which costs us the least in this example.

Let's take an example, if we want to go from A to B using the A to B game, typically we'll choose to walk or travel in a straight line. We can say that we want some kind of formula which calculates a result based on important system considerations.

Minimize Energy

Minimize Time

Maximize Safety

and so on which is the cost to the LOS/HOS

Based on inputs like this we have a formula that calculates a numeric value

Calculate payoff =

Percentage Energy Expended To Effective Currency + Time Converted to Effective Currency + Safety Converted To Effective Currency

Less energy expended means a lower currency value

Less time means a lower currency value

Higher safety means a lower currency value

Games Activated Based on Message Criteria

All games are not equal. By this I mean we can have several games in a goal but only certain ones are activated depending upon the message types that it is receiving. Therefore several games compete to calculate their payoff depending on the input message types.

Let's take an example of the robot in the nuclear reactor trying to go from A to B

Message Obstruction Found

Games which match this case are

Game Drill Hole, payout is 1000

Game Alternative Route, payout is 50

Game Climb Over, payout = 10 ← in this case, this requires the least work for the LOS

Games which do no match this are

Walk In Straight Line – this does not accept Obstruction Messaging

This way we can consider different cases by means of Criteria filtering and generating. It's likely that the Walk in a straight line game generates an Obstruction message when it cannot move any further.

Improving Games And Goals

Using an approach like this we might enter a robot competition for example to shut down the nuclear reactor and the robot may make it into the building but get stuck at a certain point. Maybe the robot detects a certain type of obstacle but creates a game that does not work. How can we fix this?

In the simplest case, we can create the situations for a new game and assign it a higher payoff, then update the Games and Goals. We would typically call this patching software.

In human behavior however, we are able to create our own new game when the existing games fail. This is a game called **Adapt And Create New Game** which from its name creates a new one. This is what we have evolved over millions of years. This would require a separate chapter but the concept is important. An advanced thinking machine design should be able to create this.

Messages Containing Data

A message declaring an obstacle on its own is not sufficient information for an A to B game. A message needs attributes.

Obstacle Message

Location x,y,z

Object type

Object size

Object weight

etc;

End Obstacle Message

This informs games which might specialize in large objects versus those that specialize in small ones. In such a case, it might be simpler to use Game Hop Over for example. The detail of of the obstacle need to be detected by Sensors.

Payoff Versus Wisdom

In the famous case of Solomon who had two women who claimed to be the mother of a child, Solomon came up with a wise solution to this problem which found out who was the true mother. The Game payoff was applied to the mothers as opposed to the child but this requires a great deal of human intuition to produce this elegant game. Therefore it's suffice to say that not all payoffs will fit every solution. There will always be exception and the Payoff mechanism must be aware that exceptions exists where the designer did not expect this case. This is where a Wisdom Operator is required and is a separate discussion.

Determining The Sequence of Operators

Using this design pattern we need to be able to define the sequence of Operators. At each level, we know the next possible Operators based on the messages which are produced and the data they contain. Some are fly wheel operators and can loop and know how many times and when to stop. The sequence of operators form the equivalent of a programming language. There are 'if then' operators which operate on Boolean input messages and so on. The main difference is that the next operator is the one which matches some criteria similar to a game so there may be more than one next step so to speak. Photosynthesis for example is a fly wheel operator which does several steps before starting again and knows when to stop. An output message for one operator running a game becomes the input message for the next operator.

Operator if-then

if true call operator with best possible match to message (Message Loop Data for Operator A, Message Finish Up for Operator B)

Operator A Fly Wheel loop until criteria satisfied

Operator B Exit Game and Generate next message.

So we have next possible games and next possible operators-states based on criteria and payoff.

What is the difference between Games and Operators? The games define the high level semantic goal like Go From A to B and Shut Down The Reactor Game and cover the Action Space and so on. The Operators inside the games can contain the game type like Press Button or Turn Wheel and so on. They perform actions which are either virtual or physical. However for the most part operators are games but some are very simple and do not require all the features of a full game.

Goal Shut Down Reactor

Games:

Game Go From A to B

Game Shut Down the machine

Game Go From A to B

next options based on message criteria

Clear Message → Go In A Straight Line Game

Blocked Message → Climb Ladder Game

Go In A Straight Line

Operation Fly Wheel Start at A and Finish at B

    Operation Move Feet

    etc;

So we're dealing with modeling the game types, the message types, the operators that the game types use, the trust, the personality type, the Bayesian trust and potentially even the gender. **An operator is itself a game type but a lower level game that may talk directly to hardware or muscle tissue**. A body for example is a high level operator containing many layered games. Additionally an operator may be a game type which performs like a traditional programming language with loops and if-thens. **Basically everything is a game and everything is an operator and the level of sophistication depends on the situation.**

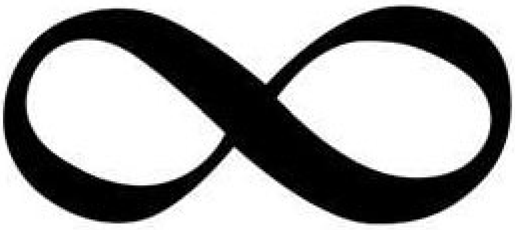
Let's talk about the game types next.

Chapter 22 Game Types And Game Theory

In this chapter I discuss Game Types which form Game Theory.

Defining An Infinity Game

Let's say we have a need to create life as we know it. If we use an infinity game where all we have is replicate and refine it should be theoretically possible. There are two goals Replicate and Refine and each has a respective game Replicate Game and Refine Game. Both games interact with one another infinitely.



Each side represents a single Goal or Game. The output of one game is the input of another game. Both games want to maximize their payoff. Replicate wants to have plenty of food/energy to replicate and some instructions. Refine seeks out ways to find a good food/energy source and contains the blue-print for what the overall game looks like. Refine just changes the design but does not do the building. Replicate will just follow the design that Refine gives it. They will do this infinitely. Replicate can copy itself so there can be many copies of this replicate-refine design and over time they are all trying to improve their design. They don't necessarily need to know about the other off springs. Before one knows it there are billions of these Infinity Games running with lots of shapes and forms. Refine keeps trying all sorts of ideas (shapes, forms, strategies etc;) to find new food and once replicate has enough food thanks to refine and is safe then life will evolve into many different shapes and forms.

We can also apply this design pattern to the known Universe. It starts at a point where there is nothing. Replicate builds the universe and refine changes atoms and builds planets and then us. According to this theoretical idea, these two forces are feeding on something we do not see to give it energy and an instruction set somewhere possibly inside the wave function. Maybe these are some kind of tiny waves which give us our energy possibly.

So an Infinity Game is possibly one of the most powerful games that one can create using this idea. All you need are two goals and two games and let it run! Our DNA is one of the possible solutions to this game where the instructions for replication are contained there and the replication operators take the instruction set and build us.

Replication Game

Process Instruction Set from Refinement

Build One or more Objects

Use the food/energy available

Copy ones self if required

Send the Instructions back to Refinement

Inform the Refinement Game if it had enough Food/Energy

Refinement Game

Receive the Instruction Set

Measure the amount of food/energy available

If the payoff is running low alter the Instruction Set

Send the altered instruction set back to the Replication Game

Do this Forever or Until the cycle completes.

The instructions can maps to DNA, the enzymes which process the instructions are similar to the replicators and the refinement is calculated by the HOS/LOS using epigenetic markers and the like.

A cell might need oxygen and water but a human being might need a good paying job providing safe accommodation and a steady food supply but ultimately we are dealing with the same basic needs in a more sophisticated setup. Replication is having a family or a cell dividing.

Life never stops moving forward but an individual plant or being may die. However the ecosystem is always moving forward infinitely. Machines are already following this life-cycle as people develop new smart-phone models for example and binning their old one. This requires human intervention at present to improve the design and manufacture the new models (replication and refinement) but people use these devices because they help the A to B game for example providing trusted information on routes and their safety with improved payouts for the games we use to move around. This is one of its many uses.

A game like this can be implemented by means of two static functions passing the necessary information between them.

Intelligent Design versus Random Accidents and Darwinism

In this design the start of life is modeled as an Infinity game. The important point to make here is that there is a payoff calculation between the games. In this theory, this constitutes intelligent design. By this I mean that the game actually has to do a calculation and realize that the infinite game is not working and alter itself using a change to specialization. This is not just some mindless mechanism that occasionally does a "random" mutation mindlessly as Darwin suggests. This is a self-aware game that's trying to improve itself right form the get-go. Therefore this is how this theory sees life and the Universe itself as some kind of Infinity game that's self aware, always seeking self-improvement. This is an important point to make in this document and not just some philosophical idea. The design pattern really needs this self-awareness to succeed. Start small, start smart.

A Human As Replicate Refinement

If we apply the Replicate/Refine design pattern to a human cell then the nucleus is Replicate. This handles cell division and the like and contains the running master program. The surrounding Cytosol is the Refine where the program runs and interacts with the outside world and monitors how it is performing and determines if adjustments are required which are sent back to the nucleus. Epigenetic markers are like payoff functions measuring the state of the system.

Dynamic Games versus Static Games and True AI

A static game is one that we can program into an AI as some kind of game to help it do predict planning. For example with the A to B game, we can write the steps for the 'walk in a straight line'. However for the system to be truly intelligent, it needs to have Dynamic Games and this is what we have. By this I mean, sometimes all the games we have programmed fail and we must try new ones. Examples are Apollo 13 when the CO2 scrubbers failed. Rather than just consign the astronauts to an untimely end, the Nasa engineers created a new CO2 scrubber from components in the module and it worked.

GOAL CREATE CO2 SRUBBER

DYNAMIC GAME CO2 SCRUBBER

COMPONENTS

A, B, C, D

DETERMINE GAME STEPS

So in order to create a dynamic game we need a new goal and a new dynamic game. We need some input components which are lego like and we need to see if we can piece them together to make a CO2 scrubber.

We can imagine a robot in a nuclear reactor where it's trapped and the goals have failed and it needs to find out what is around it to create a new game. To a certain extent this is 'true AI' and this is what I'll call a Dynamic Game. Now, you may not want your carpet cleaner to have dynamic game abilities but you may certainly want your new an improved Planetary Explorer Robot to have it for cases that one could not forsee!

Once the dynamic game is created, it can be refined.

Refinement is a form of dynamic game but the point is that the game needs to exist in the first place so using this idea, we can imagine that an initial game might be just one or two operations. Over time it becomes many more but we also need to support creating games from scratch.

Clearly this requires some kind of oversight so that the games are beneficial.

Games Which Dynamically Tender For A Game Solution Set

Using this design pattern, a Game is called from a Goal and the Game runs a program which enters a **tendering process for a solution action path**. The first part of the action path is to calculate the virtual cost of the tendering process. Only games which match the criteria are considered for the tendering process. The physical solution (movement) is only implemented when the virtual tendering solution is generated. The solution is constantly checked to see if it can be optimized further.

Game A to B

Criteria Raining

Criteria Sloping

Criteria Mountain

Game Mountain Cimb

Supports Criteria

Raining, Sloping, Mountain

Predicted Cost, Predicted Payout

Risk of failure 20%

Game Walk String Ladder

Supports Criteria

Raining, Sloping, Mountain

Predicted Cost, Predicted Payout

Risk of failure 60%

The A to B game enters the tendering process with the two games which match the criteria. They return their payoff and the effective currency criteria is calculated and one of the games is chosen. Each of the calculated virtual steps is calculated and this forms the game plan.

Then the game plan is executed by actual steps and the effective payout is calculated. If it differs markedly from the estimation, then the game request is re-run to see if an alternative is possible.

This is very much like when a person is going somewhere in one direction and they suddenly stop and turn around when they calculate that the plan they have for A to B is not optimal. So we need to keep calculating the game plan to see if it needs to change every few steps.

So we can have

Related sub-games which are either static of dynamic

If no solution game steps or there is a recalculate, match sub-games which match the parent game criteria

Generate Tendering messages. Get predicted and payoff values. (virtual circuit)

Calculate the most optimal, there may be many steps.

Build a game plan of action steps

Execute the actual steps (physical circuit)

Monitor predicted vs actual

If is is different, then call refine and restart the tendering process

Breaking Down By Goal And Game And Sample Timing And Not Falling Over

So if we have a goal, we need to create one or more games and we also need to define the message criteria. For the A to B game, we tender and then calculate the virtual payoff. Then submit the request to the game's physical circuit.

Next we have to ask an important question which is how often do we re-run the game to check that everything is as we expect?

For a fighter jet we need to sample in milliseconds or less. If we are walking we need to do it every step. If we are pushing a pram maybe less as we can coast with the pram. If we are sitting in a wheel chair and being pushed by someone else, maybe rarely or not at all if the other person knows the way.

Another point to make is that it's up to the physical circuit and other games running to generate any important feedback criteria messages which may change the game.

OBSTACLE FOUND

FALLEN OVER

SLIPPED ON ICE

TWISTED ANKLE

The foot must inform the A to B game of the type of terrain.

FLAT SURFACE

RUBBLE DETECTED

PEBBLES

Or if this is a fighter jet.

ENEMY PLANE DETECTED

Or if this is a car.

PEDESTRIAN CROSSING

This then feeds back into the game recalculation to find the game appropriate to the situation.

CLIMB OVER RUBBLE GAME

This means we have a very dynamic tendering process based on a rapidly changing situation. Ideally we'd have all the static games programmed into the system. However, one might want dynamic games for an advanced AI. Typically, one could imagine programmers creating the static games based on game analysis. These games can be developed independently and unit tested on their own by automation suites.

In some cases we need very fast response times which mean that we do not want to go through the SC. In this case, we need to locate the static games close to a critical motor function such as a knee or a foot where the job is just to keep the person or robot upright. This is the well known reflex test where the doctor taps the patient on the knee and it jumps up. **Using this theory we program the Goal “don't fall over” game right inside the knee and the foot for example to prevent falling over.**

Mechanism Design, Dynamic Games And AI

Another name for a dynamic game is to use Mechanism Design. In this use case, we have a payoff function we wish to maximize and we have some criteria messaging. From this we design the game. An example which is used is the idea of an auction for an item where the price is not known but we want to maximize the price. So a game is needed. In this cases, a second price auction is used where the bidder can reveal the true value of the item to them but only pay the second highest price.

Let's generalize this idea and turn anything into an effective currency. In this case, we crash on a desert island and we need to find water to survive. The water is turned into an effective currency which we want to maximize. We don't know how or where to find water so we need to figure out one or more games to satiate our thirst.

We have

TREES

ISLAND

PATHWAYS

We can use an EXPLORE game and we can produce two games.

GAME DRINK FROM LEAF WHEN RAINING

or

GAME DRINK FROM RIVER

The river game wins out but we need to find it by exploring. Typically human cities are based on or near rivers because of this as a steady source of water is required for our survival. Drinking water from leaves requires rain which we cannot depend on. Early civilizations would store rainwater in some kind of tanks.

Next we can consider the idea of games competing or goals competing in one or more individuals.

Dynamic Games And Pleasure And Pain

If we assume a design where there are dynamic games then one is not always certain of the outcome of a game. For example, one might reach into a fire and take out a hot coal. This will damage the body, so we need two new games which we all know very well. These games are Pleasure And Pain.

The Pain Game is called when the LOS suffers some form of injury or disability. This will send message criteria of damage of some kind which is a lot of work for the LOS to fix. Maybe it could be that the LOS cannot fix it. Therefore the pain game is going to be made pretty awful for the SC. The SC will remember this and not do it again if it was in bad judgment.

Pain is manufactured for the SC and pretty much shuts down all other games. For example if you experience a spinal injury the pain will be so repetitive and dreadful that you will allow a surgeon to operate on your spine which is a pretty dangerous thing to do just to get pain relief! So the pain disrupts all your normal goals and is a form of Please Fix Me! Message.

```
GAME PAIN
CRITERIA DAMAGED MESSAGING
SHUT DOWN OTHER GAMES
SHUT DOWN OTHER GOALS
DISRUPT SC
```

On the other hand we have the Pleasure Game which is a way of rewarding the SC for doing a good job. Typically, these are message criteria which indicate that game payoff is good. We have message carriers like Dopamine for this. In some cases, people take Dopamine to get a fake high but the Pleasure Game checks the frequency of messaging and throttles back the reward if the message rates are too high, shutting down and the person goes “Cold Turkey”.

```
GAME PLEASURE
CRITERIA DOPAMINE….
CHECK FREQUENCY
IF NORMAL, FEEL GOOD : SHUT DOWN STRESS GAMES
IF TOO FREQUENT, DISABLE GAME FOR A WHILE
```

Note: Stress messaging can be generated where a Game is failing to meet its pay-off constantly.

There is no reason a thinking machine should not feel pleasure or pain but if it is mostly running static games there is probably no need but it's up to the designer. But, no pun intended, it's important not to be cruel to the thinking machine.

Similarly a thinking machine like this should have an Empathy Game where it understands things like facial expressions so that it help out anyone it is working with and in need, if that is required.

Tracking The Frequency Of Messages

In many cases I have pointed out before the frequency of the events is important to the Game. There is the example of the ant which is a cleaner but counts the frequency of ants who gather food. If they reach a certain number then the cleaner can become a gatherer of food and change its type as this has a higher payout for the nest. Also, a car which is reversing and that has a sensor can stop when the frequency of sounds reaches a certain frequency otherwise the car continues to reverse. This is a safety payout.

Game Change to Food Gatherer

Criteria: Ant With Food Message Frequency > 10 per minute

Change type to food gatherer

Game Clean Up

Type Cleaner

Clean up entrance

Also, the car examples

Game Car Reverse

Criteria: Reverse sensor frequency is in safe limit

Move backwards

Game Stop Reversing

Criteria: Reverse sensor frequency is not in safe limit

Stop reversing

Place car in neutral

Chapter 23 Competition Between Individuals/SCs using Games And Goals

In this chapter I will cover how we compete with one another. We can co-operate or we can conflict. We have the idea of threat and revenge. We can also disagree. Let's discuss how this can be modeled using these ideas. For these cases we can use Interaction Games.

Defining An Interaction Game

So far we have just talked about sensing an environment and learning from it and adapting to it. In some ways, this is a moderately static environment for example like dealing with sunlight or noise or images. However, in the real world we need to deal with other beings with their own Games and Goals who may or may not compete with you. In this case, we need a special category of Games called Interaction Games. These are games dealing with other lifeforms where we need to not only identify the other life form but we also need to come up with cost and payoff strategies for dealing with them. In some cases we need to co-operate or conflict. We have the concepts of revenge and war and mutually beneficial pacts. In this chapter I will try to define many of these well known situations. Using this idea, we may not be the only one on the A to B game. There may be a group of people not all who agree about the direction. Also it might be the case where we have a game that requires another player to make their move and for the other player to wait. In the video game world, we could use these Interaction Games to make the bots that attack players more intelligent where they act as a group and learn from the player who is attacking them or even help them. **So we're trying to model Games and Goals interacting with each other but have a separate SC.** Conventionally we think of this as two or more people but later I'll show it might be how an implant co-operates with an existing limb to allow a person to walk better.

Defining An Individual

An individual is a being that physically contains one or more co-located System Consciousnesses. Typically an individual may have just one SC, for example the human brain but this may not always be the case. An individual does not have to be organic. In this theory, an individual can be hardware based or even a SIM inside a computer program or even a Personality Construct.

An individual will typically have one or more Goals and Games running at the same time to achieve its objectives. A Mind can be thought of as an individual but the individual will also contain the physical and virtual body so it refers to the overall makeup of the being. We might call this a plane with AI or a car that can drive itself or a person who lives in a city or a Sim in a game or an octopus or a bird. Individuals are not all the same but they contain at least one SC with some Goals and Games.

Individuals who Predict Plan

An individuals can predict plan using estimated costs and benefits versus actual costs and benefits. I have shown this already. We assume that an individual has a memory bank of previous experiences to draw from to determine an estimated cost. In an interaction game this is also required to determine what strategy to employ based on what we typically call experience. As well as this, there is the physical self and the virtual self where actions can be preplanned and costs calculated before actually doing them.

Interaction Game

contains

Experience Look up and Store abilities

for

Predicted Cost

and later

Actual Cost

Leveling Amongst Individuals

One of the main factors in any Interaction game is the concept of leveling. In many societies this is defined by the role of one individual compared with another. We have the psychological idea that the role makes the person possibly more important. This affects the payoff for any game and range of interaction strategies. If someone has more influence over people then the co-operation payoff for this individual is higher for example because of the influence of that individual. Therefore if individuals belong to the same species they tend to do this calculation. For example, a Queen bee is more important than a worker bee and so on. Also between species one species may argue they are more important.

A human might treat an animal poorly and say: it is just an animal! By this we mean that the level of a human is higher than an animal and therefore it's permitted to eat it and not be concerned about its general well being except how it is to eat it.

So the leveling affects the permitted treatment and the overall payout. Part of this leveling might apply also to the material wealth of an individual and what influence that individual has in terms of influencing others.

The calculation is how does this affect the payoff for the individual?

Interaction Game

consider

relative level of others

influence over others

is same species or not?

rate level of species: above, below

potential wealth

role

If the levels are the same, for example when competing for the same promotion then the interaction game promotes a competition game.

A person may eat an animal. The food is the payout but they may be required to payout for the food or the keep of the animal. Also, at a same species level, the payout may be the terms and conditions of some job in which case one may be better than another.

Personality Types

I have already covered this but an important part of any interaction is the different types of personality types there are. Someone may be a higher level but the way one is approached by the other means that the person will not take orders from the other. This is a complex topic. For example a horse needs to be broken in but some horses will not be broken. Some dogs will bite their owners and some cats will remain feral. This also needs to be part of the calculation. Sometimes opposites attract but also they can conflict.

Interactions Game

considers

Personality types

Cruelty for example means that an owner gives the other individual almost zero payout for taking orders.

This shapes the reaction of one individual to another. Some individuals may not be fair in that they ensure there is a poor payout for the opposing person in the Interaction game and there is conflict. On the other hand one person may favor another and ensure that they get a good payout function and become a sponsor for that individual while another may seek to destroy an individual.

Rating individuals is a complex topic but the point is that there is a payoff function for it using this approach for both parties.

An important point to note here is that the same things do not drive all individuals so the formula for the payout function is not the same. One person says, all I care about is my salary, while another person talks about job satisfaction being important.

Private Interaction Games and Profiling

A key point to make here is the concept of Private Interactions which will be covered later. Basically, many people will not admit their personality type while belies their payout strategy because it will affect their payout. This is like when one bids for an object. Rarely if ever will one bid the highest price up front. Often by knowing a personality type one can predict a behavior and gain a competitive edge so the art of profiling a personality type give a competitive edge. This is a separate topic in itself but it is the art of reverse engineering a personality type if the person is not disposed to admitting that personality type and is a special game unto itself.

Trust In A Interaction Game

The level of trust in an interaction game is an important consideration. Depending on the level of trust one may switch from a game of high trust to low trust. A low trust game may mean conflict and a high trust game may mean co-operation.

Game Trust

Criteria High Trust

Co-operate

Game Do Not Trust

Criteria Low Trust

Conflict!

Or

Do Not Co-operate/ Permit Access

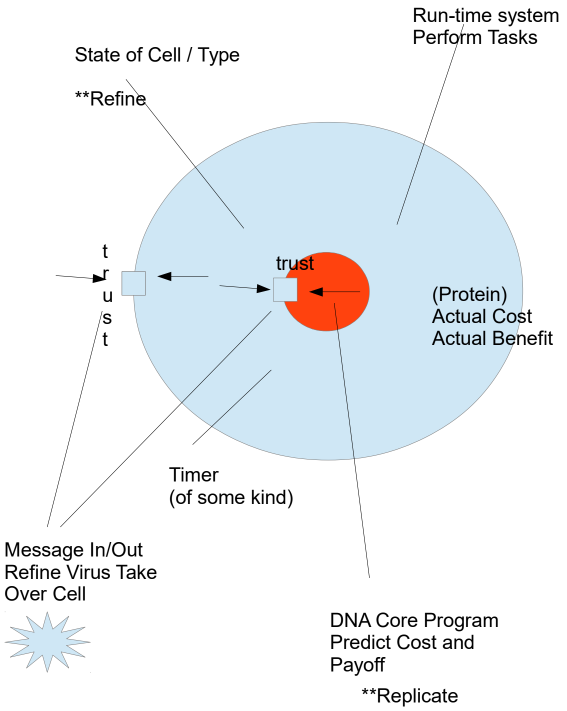
The example is an ant which forages for food and meets another ant. If it is from its nest, it will co-operate otherwise they will likely fight over the food. Therefore a Game needs to know how to trust. Typically we might know someone and build trust. Or a cell might have certain molecular keys to allow its replicate-refine pieces take in outside messages. A virus for example tries to refine its keys to break into a cell. So we can think of a Virus as the refine part or the replicate part of a cell but not both. It needs to use one or the other. The cell allows in certain protein signatures so the virus tries to emulate this but does not have the well being of the cell typically. It wants to use it to replicate itself and in doing so destroy the existing replicate-refine which keeps the body alive. Therefore the trust in this case is defined by the molecular keys. In the case of a person it might be a series of practical examples of help but in either case, the interaction is trusted.

Interaction Game

considers

Trust level with other Game

Trust is an important part of the payoff function. If an Interaction Game has trust then there is a good chance that the group will benefit and the payoff function is improved. Ants for example use small creatures to produce sugar water for them and in turn they look after them. The payoff function for both is maintained and there is trust.



Tribes Groups And Clans And Group Interaction Games

There are various names for people who group together. From the view point of this theory, in terms of Interaction Games we need to consider if the opposing game is part of a group. Team sports work in groups and supporters follow teams and share their goal to win, so an Interaction Game is grouped with other ones. **Therefore Group Interaction games cover one or more SCs which share the same payout function which is based on a group formula as opposed to an individual formula.** We see this in sporting events where the team is not only the winner but also the supporters of the team and the club.

GROUP INTERACTION GAMES

combine one or more

INTERACTION GAMES (PLAYERS)

A competitive game with two teams can be defined as two Group Interaction Games where each team is sharing the same payout function and one group is trying to maximize it according to the rules of the game.

Therefore in football one team plays the other and tries to score the most amount of points which is the payout function. The payout function varies from one game to the other.

Dynamic Leveling, Self-Esteem, Threats and Rewards

In group Interaction Games we encounter the situation where players have their own level. There is the personal esteem where the person imagines the level they are at. Then there is the level that the group think the person is at. For example, two players maybe be compared in a sport and rated at a particular level. One may be above or below the other. If a player is above they may receive sponsorship and advertisements. If a player is below then the player may be worried about being dropped from a team. Also, if these two players oppose one another this is a chance for dynamic leveling. One player can prove they are at the same level as the other. Therefore, the lesser player could be out-played or show some worth.

GAME A versus B

Criteria A < B

Foul Player! ←revenge, sent off! A!=B

GAME A versus B

Criteria A < B

Fantastic tackle! ← prove worth! A=B

Therefore the levels of the two opposing players affects the type of games that they will play when they oppose one another.

Revenge and threats are seen as an attempt of one player to another to either maintain or improve level. Different personality types will adopt different game strategies for leveling.

How To Interpret Maslow's Hierarchy of Needs And Goal Priority

How can we interpret Maslow's Hierarchy of Needs? We can map these needs to Goals. Each level maps to one or more Goals. Each Goal has a series of Games as I have discussed already. Some Goals are more important than others. For example, Food And Water is higher than Self-Actualization. So an SC needs a Goal Priority values for the Goals and Games. In other words, do I get a drink of water or do I compose a song? We can compose song if we are not dehydrated.

We check the Goals and Games in the following order.

GOAL Priority Food And Water Prio 1

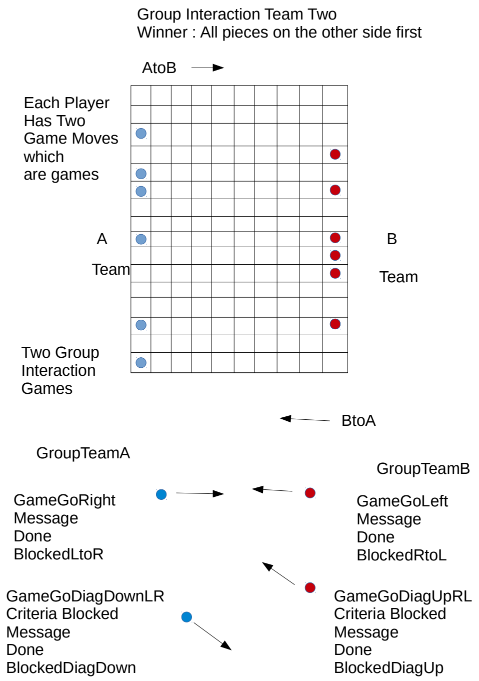
GOAL Shelter and Safety Prio 2

GOAL Priority Compose Song Prio 10

self-actualization
morality, creativity, spontaneity, acceptance, experience purpose, meaning and inner potential
self-esteem
confidence, achievement, respect of others, the need to be a unique individual
love and belonging
friendship, family, intimacy, sense of connection
safety and security
health, employment, property, family and social stability
physiological needs
breathing, food, water, shelter, clothing, sleep

Group Interaction Team Two Game

Here we have a simple game which shows the ideas so far. There are two teams on a board, team A and team B who are playing the AtoB game. One wants to go AtoB and the other wants to go BtoA. The pieces on the board can collide. In this case they have one additional move. They can either move up or move down depending on which side they are on. The team that gets all its pieces to the other side wins! The pieces appear randomly on either side but have the same number. Each piece is an actor which two movement games inside replicate-refine. When a piece is blocked, the game generates a message to indicate it is blocked. The other game filters on this message and can then try an alternative move. Each piece is independent of the other when it uses an Actor model. The programmer does not need to loop over all the pieces. Each one is a separate entity like a living player with its own game strategy which in this case is two possible moves. This is a pretty simple example but the moves and strategies can be a simple as one wants on the board or as complex as one wants. Here there are two possible moves.



Note that we have not handled the BlockedDiagUpRL or DownLR. It's up to the designer of the game how to handle this message. I have not calculated a Winner state either but I just want to focus on moving pieces having options.

The game also needs to have some mechanism for storing where the pieces are on the board. For now just understand how games like this inside replicate-refine games will generate behavior of the pieces moving from one side of the board to the other, colliding and then changing direction where required. This is a little bit like people walking on a street going in a straight line until someone is in their path. In some unfortunate cases, they may keep choosing to go in the equal and opposite directions.

Defining The Physical Board And The Score Board and Physical Games and Game Referees

We can define the board as an nxm grid. Using this design pattern we can create a series of operators whose underlying implementation is an untyped actor. **We think of these as Physical Games as they represent physical structures.** Each actor contains a grid reference and has left, right, up, down, diagonal messaging. Each actor can hold a reference to a piece and has the ability to do a move and detect a collision.

PHYSICAL GAME BOARD GRID

nxm

PHYSICAL GAME LOCATION

x,y

contains Piece?

Move left, right, up, down, diagonal

Add new piece

Can report blocked

Also, the board game itself is a shared game called Score Board and it contains a reference to each piece and knows if the pieces have reached the other side. It declares the winner when all pieces have reached the other side. **So we think of this as a Game Referee.**

GAME REFEREE

team a pieces

team b pieces

check for winner

messages, winner a! Or winner b!

Master Control Program versus A Confederation of Games

So when we think of programming with this paradigm we do not imagine one large master program with millions of lines of code. We imagine hundreds and thousands of Games and Goals which form a loose confederation of Games performing different roles. The idea is to make the games as small as possible in terms of coding for ease of use and re-usability.

The Game View

For the Two Interaction Team Game we need a Game which handles the view of the board so that a third party can see the game happening in real time. This view receives messages to build the view and to update the pieces on the view as moves are decided upon and implemented.

BOARD GAME VIEW

Messages

Create

Update a piece

Game Player A and Game Player B Taking Turns

In order for people to take individual turns with this game as opposed to having each piece moving on its own, we create two players Game Player A and Game Player B. Each player asks the pieces what their preferred move is based on their predicted payout.

The payout function can be extremely simple such as which piece is the closest to the other side and not blocked. In more complex cases the predicted payout function for a move can determine and try to predict a counter move from an opponent to see if this move can have a counter move to prevent the payout function being realized. In games like chess, the calculation of the move predicts many future moves. **However the key point to make here in this design is that it is not the job of the player to calculate the payout function for a move but the job of each piece which is itself a Game to know its best next move.** However the player can then decide which move has the best payout but the piece calculates its own payout using a predict-plan calculation. The benefit of this approach is that we are not dealing with a giant piece of code which can be hard to maintain but rather we have various games which can decide a predict-plan payout and these can be added dynamically or removed depending on their success of otherwise. So the program can evolve static or dynamic games which achieve actual payout over time.

```
PLAYER A GAME
POLL PIECES FOR HIGHEST PREDICT-PLAN PAYOUT VALUE FOR A MOVE

PIECE
MESSAGE
CALCULATE BEST MOVE
POLL GAME STRATEGIES FOR HIGHEST PAYOUT

GAME STRATEGY A

GAME STRATEGY B
```

Different Players Playing The Same Game

If we adopt this approach that the piece determines the best move then all players will make the same move. In order to mitigate this, we pass in the personality type of the individual into the game itself. So, the experience of the person and move preferences form part of the predict-plan calculations.

```
GAME STRATEGY A
USE PERSONALITY PROFILE AND EXPERIENCE OF PLAYER
PREDICT-PLAN
```

One player may have a preferred opening move and so on and have won previous games using certain strategies.

The **Reputation** of an opposing player may also affect the moves one may take playing a game as well. If the opposing player is a Chess Champion one may be forced into a very conservative gaming strategy.

Games with Perfect Information and Imperfect Information

So far we have assumed that we have a game where we know where the pieces of the opponent are. In some games this may not be the case. We can imagine a game or a situation where we know where our piece or pieces are but not the opponent. In warfare or the game battleships this can be the case. Or in a game of cards, you may not know the cards of the other players. In this case we still model the predict-plan pay-off function but you must work with any data at your disposal. In some cases, one might try an entirely random function for the payoff. If one plays the lottery and tries to predict-plan the outcome numbers, the number of solution sets is so large one can for example choose from a machine that generates random results or one may pick some personal numbers like birth date and the like. In a game like battle ships the payoff function can keep track of the previous parts of the grid which were picked and make some assumptions based on previous results. In a game of Poker for example, one must try to guess the player personality type if the player is bluffing possibly using tells. In other games players illegally count cards but the idea is to try and remove the element of chance. In warfare one may drive up a road and not know if there is an opponent waiting. However the element of chance is likely in most games like trying to find a safe place in a forest while drinking water free from predators.

```
GAME TRY TO FIND SOMETHING
RECORD PREVIOUS LOCATIONS
RANDOMLY SEARCH NEW LOCATION
```

Typically we try to find information which can be used on a more established static or dynamic game. Archeology is a good example of this where something is found in the ground and from this a broader game can be played where one tries to understand what the place, in what time and how the people or animals lived in these times.

So in the Team Two Interaction Game when player A moves the pieces, the player may not know if player B's pieces are right in front of the player. Player A must try the move and record if the piece is blocked or not. This affects the predict-plan payoff function which has less information to work with.

The Robot In The Reactor

So now we return to the robot in the reactor. Using this idea there may be more than one robot but each will use predict-plan in order to calculate its next move. Also, the robot may be in a situation where it has perfect information about its Goal or it may not. There may be blockages and so on and even something which may compete with the robot but we have not really modeled this in the initial assumption. We turn the 'Turn off the Nuclear Reactor' into a Goal with many games as I have shown here.

One of the real challenges beyond having some software to model the Goals and Games is the machine-software interface and human-machine-software interface. We can use the Goal and Game model here to in order to try and create a seamless interface between the different pieces. I'll show how this is another game type and how this can be modeled according to this design pattern.

Chapter 24 Exoskeletons, Understanding And Extending A Being and Multidimensional Goals and Games

Introduction

In this chapter, I will discuss how we deal with extending a body or providing prosthetics but mainly this chapter is about covering what are generally called Multidimensional Games. These are Games which compete against one another based on different goals and there is also more than one Game per Goal requiring millisecond or less reaction time so it can get quite complicated! So far I have only covered games running in the same goal. However, the architecture I propose deals with many Goals of different priorities running different games that may want to control the same underlying part of a body. A good example of a complex system is both balance and movement. In our minds we think of these as the same thing while we walk but in this theory they are two separate Goals with their own games but need to control the same muscles and limbs much of the time with the same Games. This is covered under the topic of **Multidimensional Goals and Games because there are many limbs to control and different Games have different requirements**.

Many robot challenges ask contestants to solve these issues and in this document, I will provide a theoretical solution for this. This topic also covers prosthetics because it's not enough to simply attach a robotic limb, it must be aware of balance and the type of movement that a person needs e.g. walking on a flat street is different from hiking and is also different from climbing stairs but one also needs to be balanced and the movement needs to be smooth. Arm movement is also similar and also requires hand-eye co-ordination which is also a Multidimensional Goal Game. So in this chapter I will dive deeper into this proposed architecture and see how it can theoretically solve some of these difficult engineering and AI problems. Using this approach I will show how one can break out the individual Goals and test out the android design pieces for the 'Save The Nuclear Reactor' Goal.

Fires Blazing, Unsteady Terrain, Stuff Falling In An Unpredictable World And A Job To Do With Time Constraints

We start off with our action space where the Android or Robot is sent into the Nuclear Reactor to shut it down. It can calculate an optimal path as I've shown but it needs to be able move and stay upright. A vent of steam may hit it unexpectedly and knock it over. Rubble may lie strewn across the floor. Wires and pipes may trip up the Android. Something may fall on it! It's a mess!! A sinkhole may open up and the robot may need to jump and land and regain its balance. If it falls over, it needs to know how to stand up again. If it falls into a hole it needs to be able to climb out. Clearly, a design which can handle all of these cases is not simple and in this theory we have evolved various Goals and Games which can co-operate to deal with these cases. It's a hard problem but let's get started!

Maintaining Balance

If one monitors early design of machines, they are pretty easy to push over. As they fall they tend to remain pretty static and components and sensors can be damaged. Over time this will change of course. We need to have different games to protect the body as we fall. Using this design we have games for maintaining balance, providing counter balance and dealing with the case where one knows that one will fall in any case and deal with it. Typically if one looks at a human being falling we tend to reach out our hands to protect our heads and neck. If we fall to the side, we stretch out an arm in that direction. Typically one can look at blooper videos for how people deal with falling in different situations. These need to be modeled into games. Filters can be applied to deal with the specific “falling situation”. These higher priority games take precedence over other games from lower priority Goals like cycle bike. So if we crash a bike we immediately use these games and stop trying to cycle a bike Game by moving our legs.

This **Maintain Balance Goal** should be running pretty much all the time as it is a Core Goal while we are standing. Typically small children have this ability early in their development as we watch them take their first steps. We can think of this as both human development and “Goal Development”. Once the ability is there, it is there for life for the most part barring any other problems.

Goal Maintain Balance Priority 1

Game Maintain Balance

Filter Standing

Use eyes

Use ear sensors

Use body sensors

Game Counter Balance

Filter Off Balance

Wave arms

Move Legs

Change Posture

Game Falling Over

Filter Falling

Sideways Falling, Use Arm

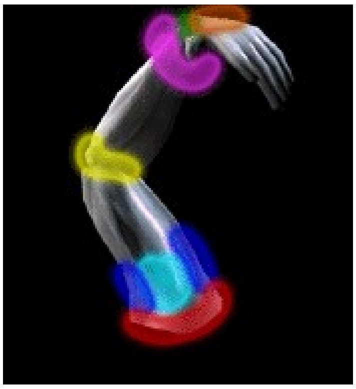
Forward Falling, Use Both Arms

Falling Backwards, Stretch Down Hands

A Body As A Three Dimensional Board Game

In order to extend the idea of Games to a thinking being with a body containing various limbs, we need to extend the idea of a flat board in two dimensions into a board game in three dimensions with level moves. The simplest analogy is to think of a three-dimensional game of chess. The different pieces represent the different movement types similar to a limb of a body and the different levels represent the connected limbs. So here we have the idea of **Connected Games** where one game affects the other at the next level and so on, similar to moving an arm causing a hand to move.

In this case the pieces are the Games which determine the types of possible moves of a limb arm/elbow/wrist/finger. A human arm for example has seven possible degrees of freedom, ranging from the shoulder to the wrist and the fingers.



We can think of this as a Chess board with seven levels and where pieces at each level are related to games at each level. This is a simplistic view but this is the idea to first understand that there are seven levels of connected Games. In reality there are many muscles and tendons as well but suffice to say there are seven degrees of freedom.

Next let's look at the Virtual Circuit and the Physical Circuit in a human being and how we can model thinking about moving before actually moving using this theory.

Game Shoulder

Connects to Game Elbow

Position x,y,z

Rotation x,y,z

Game Elbow

Connects to Arm

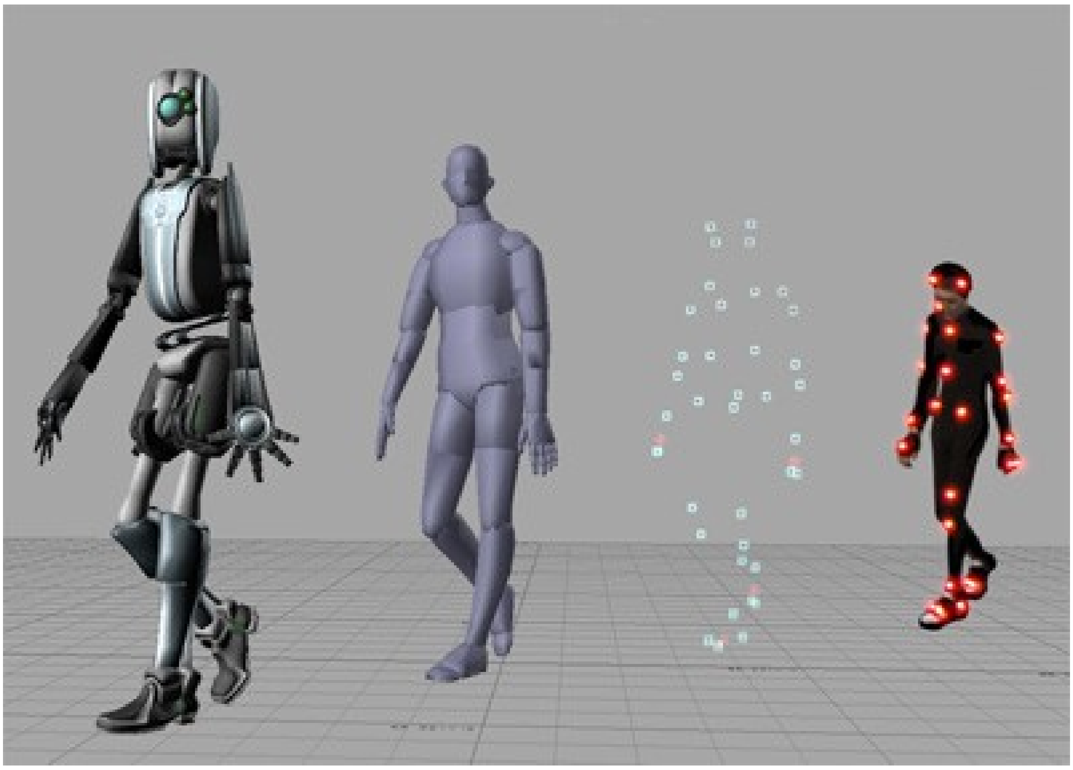
etc;

Motion Capture Suits and The Virtual Circuitry

Today, many movies use Motion Capture Suits where actors perform moves and the movement is painted over with special effects. According to this theory, the human Virtual Self is formed from an internal motion capture system using body sensors which builds our sense of self. This is a combination of the sensors in our body which form the Virtual Self which is a series of Connected Games that tells us about our body position in three-dimensional space and what our options are depending on the needs.

Our Virtual Self does not know exactly how we look. For this we need mirrors and the like. However, it does know how we are connected together and what possible moves are open to our bodies based on our position and posture.

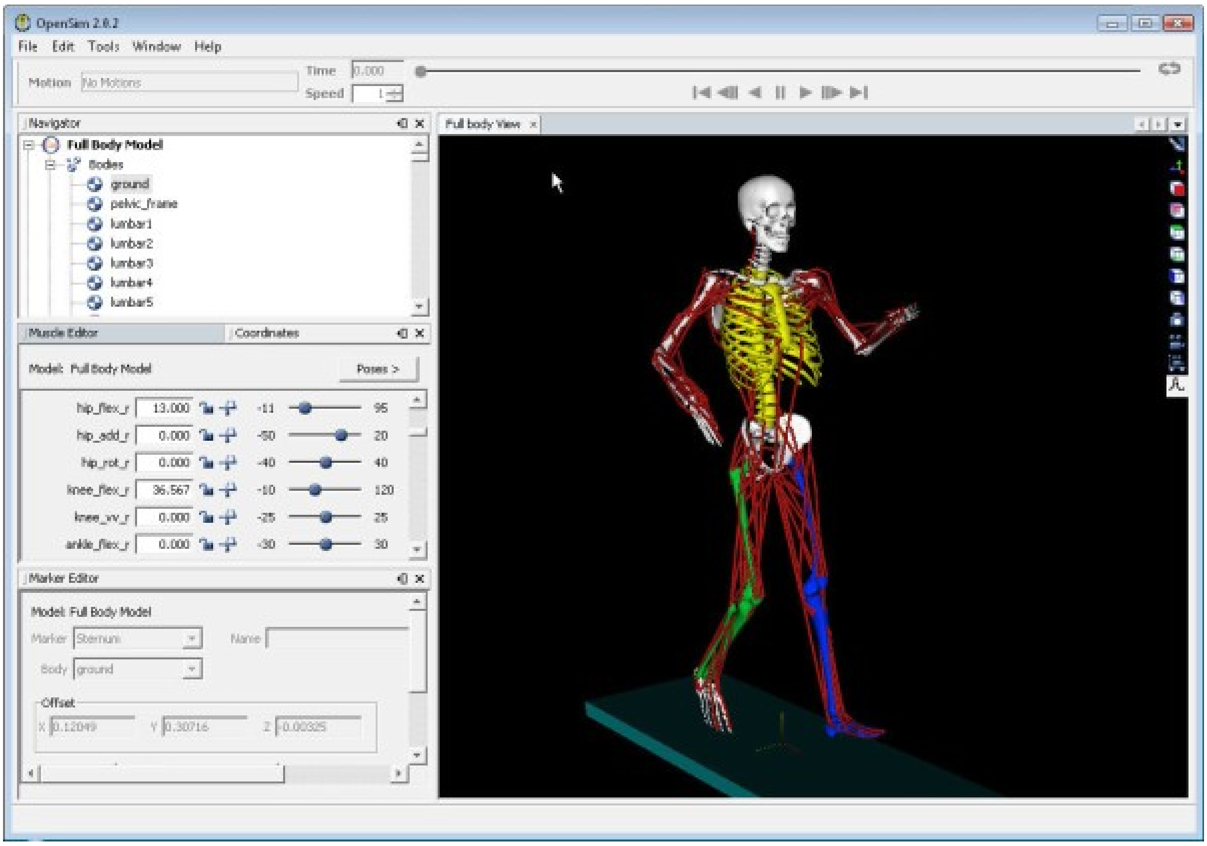
For example, we can close our eyes and imagine our Virtual Self moving our hands without actually doing it. **According to this idea, this is us applying Virtual Games to the Virtual Self for Predict Planning.**



Bio-mechanics And The Virtual Self

In order to accurately model our Virtual Self if we were a Machine trying to shut a nuclear reactor, we would like very accurate information on what strength our body has and how it fits together. For this we can use an internal program that models living beings using Bio-mechanical principles. One such program at the time of writing is OpenSim. However it does not have to be this one but I will use this one for the sake of example.

What we need is a piece of software which can model the Virtual Self accurately and try Virtual moves. What we need to do is apply Connected Games to the Software Bio-mechanical model which can determine what the best possible payout is for a certain move. For example, if we are in the air and we are dropping to the ground, what is the best way to fall to minimize damage to the body/feet/ankle. To calculate this we need to know the strengths of things like ankle joints, feet, toes and knees (for example) and what the force of the collisions will be. For example, if a Robot pushes out its hands and the collision force is too strong, then the Bio-mechanical program can detect this as it correctly models the strengths or otherwise of the body pieces and report the effect if this move was undertaken. This prevents the robot's hand being broken as it adopts Tuck and Roll for example.



Motion: No Motions
Time: 0.000
Speed: 1

Model: Full Body Model

hip_flex_r	13.000	-11	95
hip_add_r	0.000	-50	20
hip_rot_r	0.000	-40	40
knee_flex_r	36.567	-10	120
knee_xv_r	0.000	-25	25
ankle_flex_r	0.000	-30	30

Model: Full Body Model
Marker: Sternum
Name:
Body: ground
Offset
X 0.12049 Y 0.30716 Z -0.00325

Ideally we would like a View of a being like this forming the Virtual Self which is accurate. Typically though in humans we do not always know the strengths of our bodies but a machine with a Virtual Self like this where the Connected Games have this Biomechanical information means that the Game they choose for a certain move has some accuracy. This would be helpful for Shutting Down The Nuclear Reactor.

The Physical Motor Circuit

Once the Game Predict Plans the optimal choice of movement is picked. This choice is then sent to the Physical Games which receive a Movement message. This dictates what direction the body should move in. I will talk some more about this in the chapter relating to

Advanced Predict Planning which covers hand-eye co-ordination. For now, we can just assume that the Predict Plan figures out the next physical movement to be implemented.

Prosthesis, Gateway Games And The Honest Broker

One of the goals of this design is to produce a Prosthetic design which is intelligent and makes the life of the user much improved. In earlier discussions, I talked about the SC which has goals and games which predict-plan and then send the results to a motor circuit for example. However, what happens when a person suffers some form of disability? They may lose a limb for example or may have some congenital defect. In such a case, the predict-plan games in the brain are fine but there is no physical limb or limbs to honor the motor requests and the person may have limited mobility. What can we do here? Traditionally a prosthetic mirrors the same and size of the missing limb but what is not present is an Intelligent Design which contains Games and Goals. So what do we need?

What we need is an Intelligent limb that contains its own SC. This SC must have its own dedicated Bio-mechanical data with which it can predict-plan. It must also have Goals related to its operation. For example, for a leg limb it needs a Balance Goal and a Movement Goal which determines the type of movement required based on messages from the floor for example. Also, it needs Gateway Games. **A Gateway Game acts as an honest broker between the prosthetic limb and the connection points to the person in question.** In

some cases surgeons may move nerves or attach the limb to particular muscles which the person can flex and move. So the Gateway Goal handles the human-machine interface.

So an intelligent leg prosthesis should ideally have

1. Its own SC
2. Its own Goals
3. Gateway Goal
4. Balance Goals
5. Specific Movement Goals
6. Bio-mechanical data

Therefore it should be in equilibrium with the body it is attached to but also be a thinking machine that has its own goals. The limb components should be connected games and optimized to understand the needs of the user via the Gateway Goal. The prosthetic limb needs to have its own SC because it is a separate entity but also co-dependent and that is the purpose of the Gateway Goal which maps the connected user who uses some type of Interaction games.

Chapter 25 Functional Messaging And A More Complete Design

Introduction

In this chapter I will discuss something that is important and missing from the current design. As present, I've said that the messaging is the key to the intelligence of the Thinking Machine and that we can use Set Theory to work out logic and so on but the examples I have shown show the messaging as mostly being some kind of structured data. Also traditional Computer Science languages are procedural and this is typically how engineers create programs to make machines to do tasks so how can we bridge this gap to making data more intelligent? In this chapter, we'll talk about how we can Home Brew a new messaging language which is Functional but retains the semantic data and show how this can bridge the gap between Sets of data and a traditional programming language.

Functional Programming versus Procedural Languages

Computer science sprang from Math. Alan Turing was a mathematician and used his skills to formulate the Computability of Math and set us on a course to build programs using operators and the like. Over time programmers in the main settled on a style of programming in Commercial environment which we call procedural (but not all as I'll show). We have languages like C++, Java, PHP and so on. There are many like this.

```
If (something is true) then doSomeTask
```

Each line of code is either dynamically run or compiled and then executed. Everyone who programs in this space tends to pick some language that they are comfortable with and might be based on their college days but in short this is the procedural world that sprang from the instruction sets in the microprocessor world. It works pretty well so we do a lot of programming like this.

So we can imagine our thinking machine doing some task procedurally like a space probe and it will work but occasionally there will be a problem or two like:

Null Pointer Exception!

The program will crash on the probe and some engineer will send a signal to the probe and the program will be reset and maybe a code fix is applied and all is good until the next problem. Maybe the next one is that too many Threads are running and the program slows down because three things happen at once like a solar flare and a motor needs to move a sensor and a new calculation is required. This could be like the robot in the reactor having to deal with a lot of stuff at the same time. However, one thing we do know is: *People do not get Null Pointer exceptions!* Maybe people have nervous breakdowns but the software in their bodies does not crash and require rebooting! So we are stuck, or are we?

However all is not lost, the Mathematicians who originally designed computer languages had and still have another *better* approach which is called Functional Programming. In this language which is based on Category Theory which is an extension of Set Theory there are no null pointers and data is encapsulated in the programming language in a different way but one that is not too different. So this is the design approach we'll use for the Functional Messaging approach. Using this approach, our robot entering the nuclear reactor will not get a null pointer exception but rather an empty set of information and if the language is implemented correctly it will maximize usage of the number of cores at its disposal from a Threading viewpoint. Let's call this functional messaging language "Home Brew" where the idea is like that of a Craft Beer. There are many beers (computer languages) out there but in this case we'll create Home Brew a Functional Messaging language. I will provide some examples of how I think this could work. Clearly, these can be implemented concretely in other pure or functional supported language but over time these languages themselves will evolve so I'll stick to some examples that makes sense here providing a concrete example but generic enough for reuse.

Some Example Languages

At the time of writing, these are the types/flavors of languages out there.

Procedural

Java(compiled), C++(compiled), PHP(scripting), Python(scripting) etc;

Pure Functional

Haskel(scripting), Clojure, Lisp

Hybrid: Functional Supported Procedural

Scala(compiled), Java, C#

So the idea is to take your pick or maybe create your own concrete language if that is your cup of tea!

The other key idea is that languages are compiled into another form like Java Bytecode or dynamically executed.

Product Requirements For Home Brew

Home Brew is a dynamically generated functional messaging language that is based on Category Theory and is mathematically complete and which operates on Goals and Games at all levels of the Thinking Machine's Operating System.

Or to phrase it another way: It's not some hacky language that we managed to pull together and get working and has no real basis in Math. (However, this does not mean that it could not be implemented by patching an existing language but lets aim for the high ground.)

Earlier in the document I talked about the A.OS for the Android Operating System but it could be any kind of thinking device running static and/or dynamic games.

Category Theory As A Spring Board For Functional Messages

Category Theory is a whole branch of Math so the idea is not to explain all of it. However, I will explain the basic ideas. Basically, we have some categories that are joined by arrows which are like function operations or functors.

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[dr, "g \circ f"'] & Y \arrow[d, "g"] \\
 & Z
\end{tikzcd}$$

From this a formal Math notation can be determined.

Schematic representation of a category with objects *X*, *Y*, *Z* and morphisms *f*, *g*, $g \circ f$.

The notation allows Composability and morphisms which means that the category data can be transformed in some way by the arrows.

The idea in this document is to see the Categories containing the Messages I have outlined so far. The Message Algebra becomes Functional Messaging which is defined at its simplest level as $g \circ f$. However, as I will show, the idea of this approach is to create **pipelines of operations.** The nice thing about this approach is that if we build a language off this approach, it is mathematically complete and something both Computer Science and Math folks have a reasonably good handle on. Better still this language will not suffer from Null Pointer Exceptions which I will show and the code should be readable.

So we're heading for Functional Messages being passed around a Thinking Machine. Therefore, if you feel the sun on your skin, the idea is that your skin generates some kind of functional messages which are passed through the categories which in this case are cells in a body or operators in a thinking machine to the correct location. The message contains some kind of logic associated with what is required to do with the Functional Message beyond just containing the data. Please understand this concept before proceeding. Also it might be a good idea to read up on Category Theory and Functional Programming independent of this document before proceeding if this is new to you as this is what I will build on. I would recommend this.

The Categories connect to one another with these arrows to form a topology which in this case is the Thinking Machine in question.

Home Brew Pipeline of Operations

So let's take a practical example of how Home Brew Functional Messaging could work. The idea is that Operators can generate or consume Functional Messages and they are actively passed around the body of the thinking machine from Operator to Operator. Also note that Games and Goals are built on top of the Operators, so Games will also generate their own Functional Messages. Previously I talked about message data types. Now, let's see what we can achieve with some Home Brew style functional messages!

Simple use cases:

Wind moves hair on arm

Needle jabs skin and person jumps

Wind example

hair(a).movement(light).location(b).transport.remote.sensory.sector.brain

Needle example

cell.damage.list(abcdef).reflex.reaction.location.left.arm.immediate.pull.away.thenapply.inform.remote.sensory.sector.brain.cell.damage

So we can make up messages that encode some kind of instructions and can be as complex as we want. Note how we chain together the operations to form a pipeline of operations.

This simple functional messaging approach is very powerful because we send what we want to do where the event occurred but they also contains the data relating to the event.

Other examples could be functional messaging directing orchestra apps to build a body, and the sequencing of steps.

Orchestra example

build(a).build(b).then.waituntilcomplete.then.build(c).build(d)

Or it could be the set of instructions the cones in a human eye send to the vision processing system about they type of information it is receiving and how it should be processed.

Vision example (simple)

vision.cone(a).visual.info(b).transport.to.location.visual.processing.v1.handoff.for.processing.aggregrate

These are examples of our notional functional messaging.

In functional programming there is the concept of *Currying* by Haskell Curry. In procedural programming we would create a method with several parameters and process it. Using the Currying approach we have single method operations that we can chain together which is what we do here in our examples. Please read up on this if this is unclear to you. Much of the Home Brew implementation is supported by the Operators/Cells which understand the Functional Messaging language much like the way a Java Virtual Machine knows how to turn Java Code into Bytecode or an Apache Web Server understands PHP. So using this idea, DNA contains stored functional messaging. The proteins which are passed from Cell to Cell contains data and functional messages that the Enzyme machines understand and is their inter-lingua. This is the Theory and this is what our Home Brew language is trying to emulate so we can break out next what it needs to do.

Monads and Converting Message Types In A Flow

We can imagine a situation where the human eye takes in information and combines it into a a format of raw data which needs to go through several levels of filtering. So we start with raw visual information and have to determine basic shapes, light intensity and so on. Then we can determine two dimensional shapes. Once this is done, we figure out the three dimensional shapes and add perspective. After this, we pattern match those shapes and place them in a virtual room for the virtual self. The key point to this is that we are altering the type of information as it is processed.

light.cones.all.combine.raw.visual(a).then.process.two.dimensional(b).then.process.three.dimensional(c).then.apply.perspective(d).then.pattern.match(e).then.transport.to.sc

So this means that under the hood the data is being converted from one form to another in a pipeline, a little bit like the way a car on a production line starts out as raw components and eventually turns into the consumer object we know as a car or an order in a factory as all the pieces are assembled to be sent to a consumer. In functional programming this concept is called a Monad. **Using Home Brew, we consider this to be a mechanism in a pipe line that changes data types within the flow of the message to achieve some end result.**

It could also be audio visual information where two data types become one by merging audio and visual.

sound.(a).vision(b).then.merge.to.audio.visual(c).then.transport.to.sc

Evolution And Chaining Messages Together

Note that in this design pattern we imagine Home Brew Messaging can produce and rely on other messaging so it is not just the case that one Functional Message handles everything. In the case of an Orchestra Functional Message we can create others and wait for their result.

orchestra.doA(a).doB(b).then.wait.for.complete

doA.processC(c).process(D).then.wait.for.complete

and so on. This could be the process to build an embryo and turn it into another form. So we can imagine evolution to be a very, very long set of functional message which orchestrate in a very complex manner but the core structure is functional messaging.

embryo.grow.cell(a).reptile(b).mammal(c).primate(d).human(e)

Each of these operations like cell or reptile have many thousands or more functional messages. However if we want to evolves this we can add another operation like somethingNew(f).

embryo.grow.cell(a).reptile(b).mammal(c).primate(d).human(e).somethingNew(f)

Storing And Retrieving Home Brew Messages

We can imagine a simple mechanism where the operators in games can generate Home Brew messages dynamically but also store and retrieve existing ones where they have achieved their pay-off function. We can therefore think of Chromosomes as a storage repository for Functional Messaging.

message(A).chromosome(B).store

or

chromosome(B).retrieve.message(A)

Routing Messages

In some of the examples of Home Brew we have a routing description. This raised an important requirement for Home Brew which is where do the messages need to go to. For example, visual messaging needs to go to the V1 section of the brain. Audio visual needs the SC and so on. In the brain we have dedicated circuitry and different stop off points a bit like a subway. So the Functional Message needs to be implemented somewhere or in more than one place. If one looks at human anatomy then we have the endocrine system where it is released into the blood stream and the message is for certain parts of of the body.

release.system.wide.message(a).transport.to.liver

Therefore once the message finds liver cells the message is delivered.

Should we have a protocol in Home Brew. The answer is that Home Brew does not really care but in a concrete implementation it depends on the requirements. For example, the Internet has TCP/IP and UDP. TCP/IP means that we keep the sequence of data and UDP means we send and expect that it will be delivered. So Home Brew can state definitely deliver or not.

release.system.wide.message(a).transport.to.liver.ensure.definitely.delivered

The implementation of definitely delivered could means that there is a delivery acknowledgement.

release.system.wide.message(a).transport.to.liver.then.wait.for.acknowledgement

As we see, the point here is that the Home Brew Functional Messaging is very flexible. We leave the concrete implementation to later but this leads to a stronger more readable design. In the case of procedural designs one can get drawn into blocks of code which semantically mean what the functional messaging means.

Human Speech, Thinking And An Interlingua for Home Brew Messaging

In an earlier part of the document, I talked about human language and why we use it and how it works. Home Brew asserts that there is an interlingua for the HOSL. Therefore when we think about anything in particular we are using our bodies functional messaging.

vision.see.car.walk.over.to.object(car)

This translates to English as: I see a car and I walk over to it.

One can note in other language that they essentially contain the same information but the words, symbols and ordering of the information is different but the total contact is invariably the same. Therefore we have locale mapping which we call learning.

vision.see.car.walk.over.to.object(car).transform.to.locale(english).spoken.language

or some other country

vision.see.car.walk.over.to.object(car).transform.to.locale(japanese).written.language

Therefore one needs to define Home Brew Messaging to encompass human thinking. This means that the existing messaging defined in the document needs to be extended to including the Functional Messaging component. So, we return to the concept of a Feral child which never received any formal education. In this case, there are no locale mappings for reading and writing but the Core Functional Messaging is present.

Difference Between Functional Messaging And Object Oriented Code

There are many existing scripting languages at present and one might be inclined to think if one is from the procedural world that some of the examples I have shown for Home Brew are like certain object oriented languages like say Javascript.

```
document.getElementById("demo").innerHTML
```

In this case, we find an item on the page and get its inner HTML. So how does object orientation differ from Home Brew? The idea in Home Brew is that we are not so much interested in Objects which are instances but we are in scripting. In the above case, the Objects existing inside a runtime system which is running in some kind of threaded flow of execution.

In Home Brew, we sit a level above this, almost at the level of the way a human thinks theoretically. We imagine the type of request we want. It's independent of threading. It's independent of specific objects. It can be one or many. We can talk to one object/Operator/Game or all the objects. **Home Brew is designed to think like us.** Think of Javascript as a possible lower level implementation of Home Brew within the Operators themselves. So how would we do this in Home Brew?

We could use exactly the same code but we'd probably be in a room in the mind of a Thinking Machine which has a representation of the web page which it is looking at.

```
mind.room.browser.representation.command.document.getElementById("demo").innerHTML
```

In the thinking machine, this operation would make sense for these type of operators if they understood html and web page technology.

The key point to note is that Home Brew is independent of objects and of threads rather the information and commands which flow between them. We go back to the arrows in the Category Theory. Mostly, we are turning the arrows into some kind of functional language. **If Javascript for example is running on a single thread, then this maps to a single Catgeory with an arrow that points to itself where the arrow is represented by the Javascript code.** The Category also understand some aspects of the Functional Messaging of the arrow. Home Brew represents all the arrows and all the Categories.

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[dr, "g\circ f"'] \arrow[loop above, "1_A"] & B \arrow[d, "g"] \arrow[loop above, "1_B"] \\
& C \arrow[loop below, "1_C"]
\end{tikzcd}$$

We think of the operators/Categories which are based on an Actor model almost has having their own dedicated CPU. This may not be the case in reality but from a design viewpoint this is the case. We are just sending Functional Messages between them. Some of the operators may represent a browser containing a web-page or may not.

As I'll show, one Home Brew command can utilize the content of many Operators to determine an end result. They can do operations in a highly parallelized way and be independent of threading issues and the like.

Amassing An Army of Operators To Do Your Bidding with Home Brew

One of the powers of the human mind is its ability to do things in parallel. In modern times, we want our machines to do this as well. Patterns like map/reduce break down calculations distributing certain sub-components of a calculation into discrete steps which can be multi-threaded. So we want Home Brew to do this and pipeline the operations.

Let's say we have a stream of information which is coming into brain and we want to filter out the distinct values and sort it and identify it.

VISUAL.INFO.FILTER.BY.2D.SHAPE.DISTINCT.SORT

We can imagine the information flowing like a stream through the operators which will take the data and spread it out and filter out the 2D shapes, then arrows join up to re-aggregate the data removing duplicates and then spreading out the information again to sort it. The arrows join up again for an operator to collect the final result. We would call this kind of programming in a procedural language some kind of multi-threaded architecture. However, Home Brew will naturally use the operators at hand, spreading out tasks where it needs to and re-aggregating them as well. **One Home Brew message can use many operators in a pipeline which are like dedicated CPUs to achieve an end result.** Note that for an architecture for this to work, each dedicated part of the Thinking Machine needs an army of Operators 'in reserve' (or an ability to create them on the fly) to handle these tasks. In the human brain for vision the model we assume is that they are created before hand.

The operators in the pipeline need to handle the various parts of the message, keeping track of the piece they need to handle.

VISUAL.INFO.**FILTER.BY.2D.SHAPE**.DISTINCT.SORT = Filter uses 'n' Operators

For now, we won't get caught up in any specific concrete implementation. However, the point is that we can parallelize operations over operators that take Functional Messaging.

Extending The Actor Model To Support Home Brew

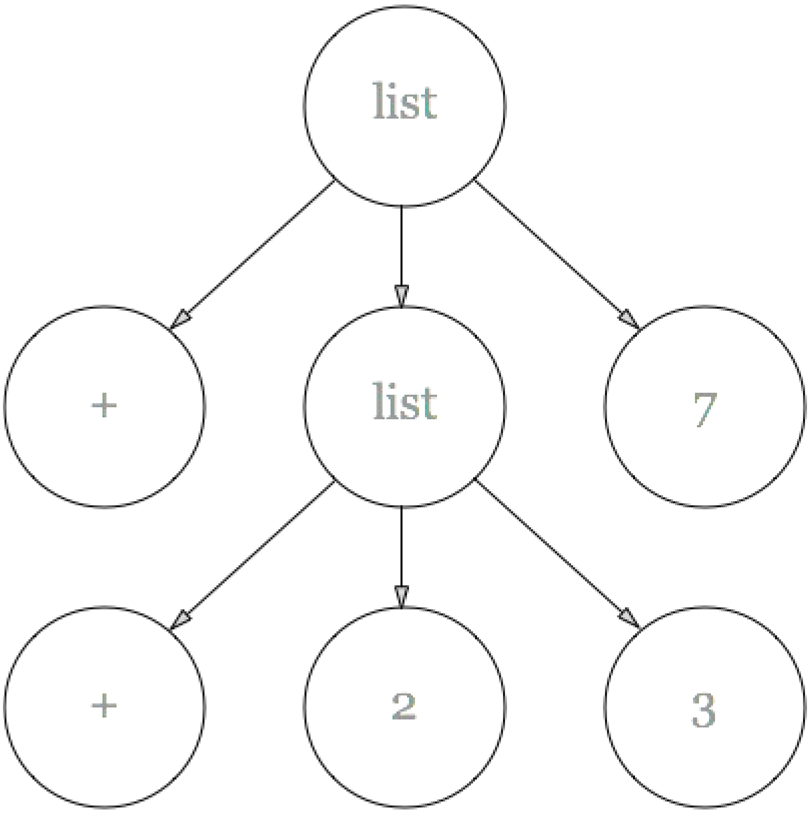
So far I have talked about the actor model which passes messages from one actor to another. Each actor has an equivalent of a mailbox which receives one or more message and sends one or more messages. In this improved design, the message is a Functional Message. This means that each actor should contain the equivalent of a Core similar to a human cell which understands the Home Brew messages. In some cases, it will generate another functional message or it might route the message to another actor or terminate at this actor and do some specific operation. In Functional Programming this type of actor is thought of as a Terminal simply meaning the end of the line for the Functional Message.

Home Brew Outsourcing All Those Annoying Parentheses

We're looking for a Craft Beer implementation of a Functional Language. We can do whatever we want that works. Yes, we really can if we want to! So we are going to outsource all those annoying parentheses which is mainly how Functional Languages are perceived as being derived from Lisp. We will work a level above the parentheses if we so choose. For example, we want to add three numbers. We have an inner operation and an outer operation. Typical Lisp notation is:

```
(+ (+ 2 3) 7)
```

This is represented by the following Operator style diagram.



So we create the Operators which are based on Actors. Each circle is an operator and each arrow contains some Functional Messaging. In this diagram this is not called out. However in Home Brew the arrows carry the Functional Messaging and there is a Pipeline of Operations. Some of the Operators are lists and others contain the operator and the number. They create a pipeline of operators which include two numbers and an operation and then add the final number. The result of this is stored in another operator which contains the result. Please note that these operators are typically immutable which is what the whole Functional Programming space is about.

So we can represent this in Home Brew in this order (or any pipeline order which contains the three parts.

math.add.2.3.then.add.7

or

add.2.3.7

or

```
add.(2).(3).(7)
```

or

```
(+ (+ 2 3) 7)
```

So we can use Lisp if we want

We can even use English

```
add.2.3.when.complete.add.7
```

It's not really my job to figure out the whole syntax in this chapter but rather to emphasize the fact that we can make this language very “English-like” or like some other spoken language. The key point is that under the hood, Home Brew is using the Operators in the diagrams or variants of this to do the actual calculation, a bit like specialized neurons in the brain. So we need lots of Specialized Operators that use Functional Messaging for stuff like Math amongst others. **The point is we don't need lots of parentheses embedded inside on another!**

Of course, one needs to map the Functional Messaging language to the operators but the point is that we can define our own mappings. It doesn't have to be Lisp notation.

For Home Brew we need to figure out the Category Theory diagram containing the Category Types and the Functional Messaging required and from this we can produce the Home Brew language as it is a combination of both. This will be covered in a different chapter of Home Brew Cookbook Examples.

Chapter 26 Parasites And Altering The SC And Functional Messaging

Introduction

In this chapter I will discuss something that is unpleasant to most people including myself and that topic is parasitic infections and how it maps to the Thinking Machine design. It's an important topic because it shows how a parasite can take over a host organism or organisms in order to reproduce itself. Sometimes the host may not even be aware it is there but in most cases there are some indications the parasites are present such as a fever and so on. I will deal with some well known parasites like malaria, toxoplasmosis and so on. Each parasite has a different strategy but I will explain how to understand them in terms of this theory where a living being has an SC and so on. This will not be an exhaustive medical description as I'm not medically trained rather a design discussion of how the parasite operates. I will also show how this parasitic design pattern can be applied to Functional Messaging for defensive purposes.

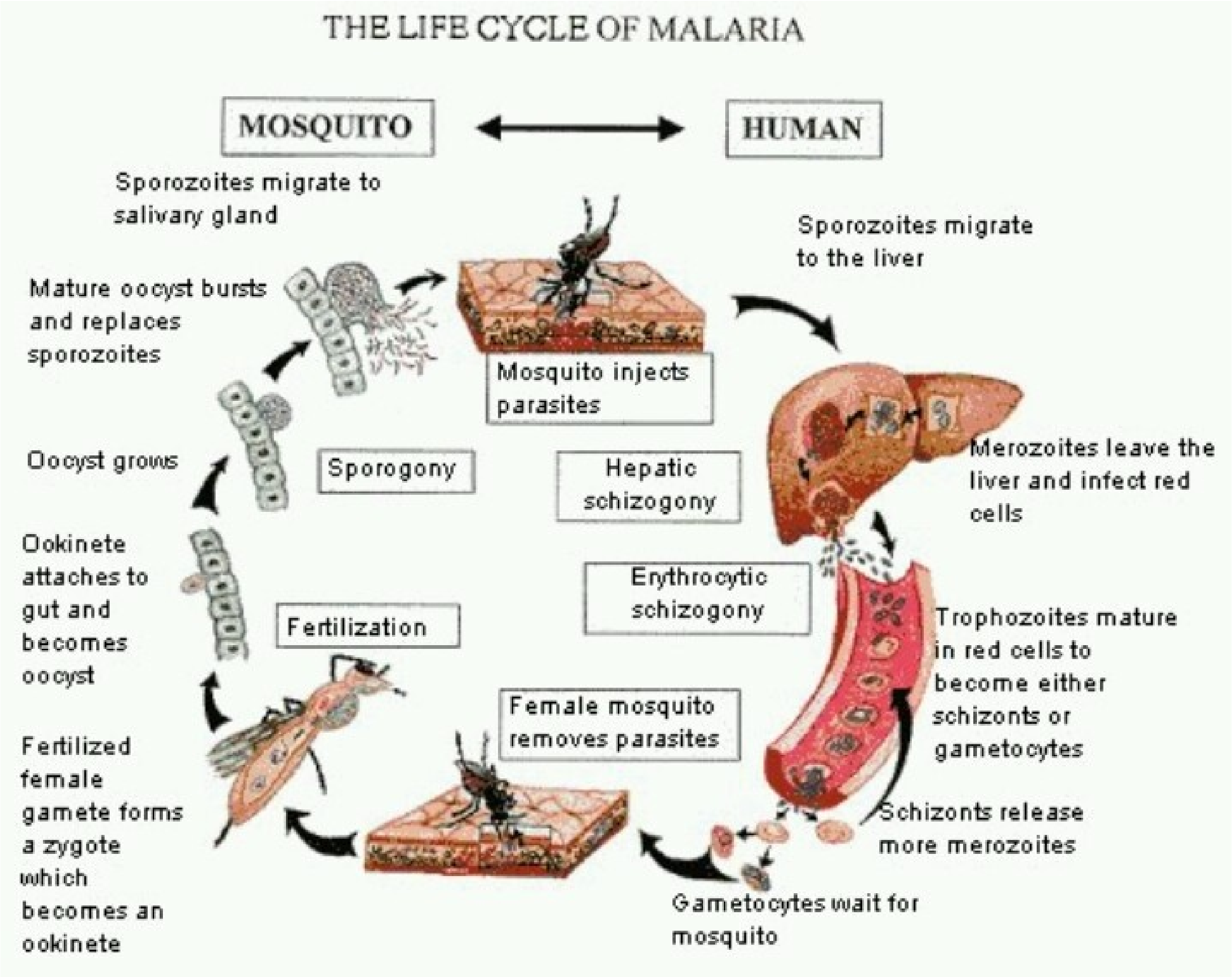
Defining A Parasite

Let's build off the Thinking Machine design. Each thinking being has an SC. Earlier I showed how an advanced prosthetic has its own SC and its own Goals and Games for a Thinking Machine design where the additional SCs try to complement one another and are in balance. A parasite is typically out of balance with its host although humans do contain bacteria which is useful so we focus on the ones which are not useful. **So a Parasite is one or more SCs which are running their own Goals and Games <u>inside</u> another host SC.** The parasite typically has intimate knowledge of its host's SC and Functional Messaging and makes use of it for its own benefit. Unlike a virus which is typically only one half of a replicate-refine cycle and requires other cells to complete its own replicate-refine, a parasite has its own replicate-refine and the ability to mimic the host in some way to achieve its own Goals which is to replicate and refine itself. Some operate at the LOS level and others at the HOS level as I will show.

Malaria

This parasite hops between two hosts to complete its life-cycle. It is an intracellular parasite where it crawls inside red blood cells in part of its life-cycle. So we can think of this as a LOS parasite attacking blood cells and the like. What the diagrams does not show is that the parasite understands the LOS and HOS code of its hosts in order to refine and replicate itself. It has the ability to read the code and encode it into its own life-cycle so it evolves with its hosts Functional Messaging. See how it produces sexual versions of itself in the human host possibly using some of the hosts messaging to keep its understanding on the hosts up to date. The infection kills older people and children typically but not adults because it needs them to reproduce to continue as its hosts. It also knows its way around its hosts by understanding the Functional Messaging relating to transportation locations and cell types. For example, it knows how to identify the liver and target it and get around the bodies immune system by hiding its presence. They key to defeating this is to understand the code in its DNA which can read the hosts LOS Functional Message by altering that ability somehow. One the other hand it would be very interesting DNA code to understand how it does it because this could be re-used positively in targeting human disease. For example, knowing how it targets the liver might be useful if someone had cancer in a particular part of the body and there was a drug to target it. One could use the malaria approach. Also, if a drug can target a cancer cell, the ability to do an intracellular insertion would be useful but then one thinks of movies like I Am Legend where humans are transformed by a supposed miracle cure.

In a thinking machine, one can think of this a little but like a LOS/HOS Toolkit which can read the Functional Messaging of a host organism and splice out bits useful to its own SC. What's interesting here is that the cells which belong to the single celled organisms have a 'defense mode' where they form a shell around themselves when they are attacked by an immune system. So in the main this is a LOS/HOS attack parasite. Highly evolved. Formidable.



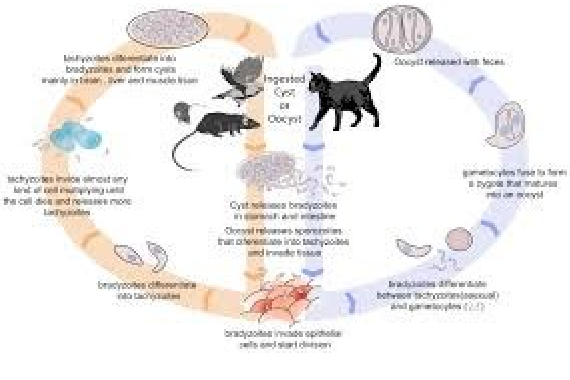
THE LIFE CYCLE OF MALARIA
MOSQUITO
HUMAN
Sporozoites migrate to salivary gland
Mature oocyst bursts and replaces sporozoites
Oocyst grows
Ookinete attaches to gut and becomes oocyst
Fertilized female gamete forms a zygote which becomes an ookinete
Sporogony
Fertilization
Mosquito injects parasites
Hepatic schizogony
Erythrocytic schizogony
Female mosquito removes parasites
Sporozoites migrate to the liver
Merozoites leave the liver and infect red cells
Trophozoites mature in red cells to become either schizonts or gametocytes
Schizonts release more merozoites
Gametocytes wait for mosquito

Toxoplasmosis

Toxoplasmosis prefers to target the brain so we are dealing with a parasite that wants to alter the choices/Goals/games the host SC makes. Typically, the parasite uses cats and rats. The infected cat passes the parasite in its feces and the rat eats it. However, rats have an aversion to cat urine and they flee. Toxo targets the part of the brain which handles the neurons and rewires its brain so that it gets a dopamine response instead of a stress response. This is a game pay-off chemical which rewards the SC for doing a good job. So the rat is no longer afraid of cats and is eaten and the whole cycle repeats. Toxo can get into human and there are some ideas that people become more risk averse.

Let's approach this from the Thinking Machine. The parasite SC also knows the host LOS/HOS and know how to target the SC by altering its pay-off function for something which would normally be life threatening to it. It builds a neuron like structure to produce dopamine. So this parasite essentially hacks the SC of the organism to get it to behave in the interests of its own SC Goals. To do this, it must understand what part of the brain to target but also hot-wire the brain and splice into it. Here the parasite becomes part of the host organism. This is OK for Toxo because the expectation is that the host organism will be eaten by the cat. How can this be useful for humans? Maybe if it can be understood how the parasite does this, it might be possible to target and regrow damaged neurons in the brain. Then again I have seen the show The Walking Dead. Brains. Need brains.

So this is another toolkit that rewires the SC Goals and Game of a running system.


tachyzoites differentiate into bradyzoites and form cysts mainly in brain, liver and muscle tissue
Ingested Cyst or Oocyst
Oocyst released with feces
tachyzoites invade almost any kind of cell multiplying until the cell dies and releases more tachyzoites
gametocytes fuse to form a zygote that matures into an oocyst
Cyst releases bradyzoites in stomach and intestine
Oocyst releases sporozoites that differentiate into tachyzoites and invade tissue
bradyzoites differentiate into tachyzoites
bradyzoites differentiate between tachyzoites(asexual) and gametocytes (♂,♀)
bradyzoites invade epithelial cells and start division

Note: There are also Zombie snails and Zombie Grasshoppers that have their SC taken over by one or more parasites. The idea is to get into the next host to complete their life-cycle but the core idea is the same replicate-refine of the parasite using one or more hosts. The defensive shell is the body of the host.

Chapter 27 Video Game Engines, Rooms And Visual Functional Scripting VFS

Introduction

In this chapter I will discuss how the thinking machine can visualize and do predict planning in a three-dimensional action space and use what I call Visual Functional Scripting. As part of this task I will take a look at video game engines and talk about how a thinking machine can use this technology with some extensions making it compatible with Category Theory and the established Thinking Machine design. I will also drill down into vision in more detail.

Imagining And Visual Functional Scripting VFS

The idea of the thinking machine is that we can build visual scripts which reflect Goals and Games and plug them together in a lego like way to solve different types of problems. Under the hood we have LOS Games with deeper code. Using this approach by using Visual Functional Scripting we can build room and the like with objects and maps. This is what we traditionally calls 'imagining'.

Recall that near the beginning of this document, I talked about a mind having a Rolodex of visual rooms and reality was the room we live in when we are awake and it reflects what we are seeing. If we are asleep we switch to another room where we dream or in other words feed Visual Functional Scripting into it to image all kinds of crazy stuff. Also when we work, we need to think and switch into a predict-plan room and architect and imagine different things to help up focus. So in this theory, the thinking machine has a form of Video Game Engine in it which knows physics and operates a Visual Functional Scripting system. This is what we are aiming for in this chapter.

Moving forwards I will call Visual Functional Scripting VFS

VFS = Visual Functional Scripting

VFS relates to objects, rooms, maps and their behaviors inside one or more rooms

The Technique of a Pick Pocket, Awareness and Focus

One might be wondering why I am going to discuss how a pick pocket works? What does this have to do with Visual Functional Scripting? The answer is not that hard to explain. We imagine an SC being flooded with Visual information defining the whole panoramic view of everything. You can see the room with a chair and a table and a person and maybe you are in a coffee shop and the pick pocket approaches. Suddenly before you know it, you are missing your wallet. How did the pick pocket do it? You were paying attention and now it's gone. The person moved by you and viola! You have no wallet. As a human being we need to focus on one or more things but our ability to focus on 'n' things is limited by our SC.

Some of us can focus on more things at once for example, imagine you are a submarine commander. You have to think about the whole submarine and its mission and all the crew. It's complicated! People get tested to see how many things they can focus on at once. So we have awareness and focus. So when you are in the coffee shop, you are aware of something, let's say your electronic device and your coffee. Occasionally you look up but in the main, you are in coffee plus gadget mode. Then a person slips by you and they took your wallet off you! It's not often a pick pocket explains their trade but they have a job to to, they need to alter your focus and over-ride your existing awareness. They need to make sure that you do not feel the wallet being taken from your pocket. In order to do this, they need to move in a way which generates another sensation and over-rides the current one and changes your focus. So this means, we can only be aware of so many things at the same time and it's possible to over-ride one with another.

I talked about this already in the document under the topic of acupuncture and how a needle in one spot generates messaging which can over-ride pain for example. So the pick pocket may gently move in a way which sends a message to the SC so that the other message of the wallet being taken is ignored. This is the skill of the pick pocket.

Typically one sees the pick pocket putting a hand on a shoulder inside a person's pivate space and that competes against an unseen hand taking a wallet.

This is a bit like two messages coming through a queue at the same time and you can only choose to process one. I showed an example of this with the sound and pictures arriving at the same time and how people miscounted. In this case, we can choose one or the other.

High Priority

hand.on.shoulder

focus.on.hand.look

Lower Priority

wallet.removal.unseen <-- missed message due to fact focus is on hand on shoulder

The key point to this section is that people have limited awareness and focus but some have more than others. **So awareness is related to the total number of messages that one can experience at the same time.** Focus on the other hand is the messages from a stream that one chooses to pick out of the stream. This is like listening to music. Some people focus on the lyrics while others are interested in the melodies.

person.sc.focus.music.melodies()

A great composer on the other hand can focus on all aspects at once.

Foveating And Focusing

Imagine the scene, a person is hanging one handed from a mountain and who looks up at the mountain face. Typically climbers puts some powder on their hands and they look for the next grip point. In this case, they are foveating on the mountain for a hand hold. This is a key ability in vision where we can zoom in on visual detail. To imagine this, one can see the mountain but the mind is shining a foveating light a bit like a torch on the surface of the mountain calculating how useful the next point is. So what happens when we do not have this ability?

This is where we enter the world of autism and a person with autism walking along a busy road. Everything is loud and moving. The person is getting information from every direction and finds the whole experience debilitating. It leads to a reality where one cannot focus if the ability is missing. One can become quickly over-whelmed and feel flooded with VFS.

A simple analogy in reverse is a horror movie where a person holds a torch in a dark room. This is an extreme example of foveating where we are in the dark but the torch light acts as our expression to see some kind of detail on the scene. Typically we find this scary because we can see so little to focus on. In a non-scary case we can see the rest of the room and pick the spot to focus on. Our visual system likes this because of the spread of visual information and we do not feel constrained. This mental focusing is related to some kind of goal and related game, for example like find the remote control where you scan each part of the room.

sc.vision.foveate.room.find.the.remote.control

sc.vision.foveate.mountain.find.the.next.hand.hold

It's also used for hunting where a human being is tracking prey. We need to be able to foveate to find the animal we are tracking, zooming in on different parts of the scene.

sc.vision.foveate.track.prey

So in a Thinking Machine example, we need to be able to focus on certain aspects of the Visual Functional Scripting to drill down into some detail. A good movie example of this is the movie Blade Runner where Deckard examines a photo with a program which successively zooms in on different part of the photo for clues.

This ability in conjunction with our bodies' motor functions leads to hand-eye co-ordination. So one of the aspects of vision we need is to foveate. In our human eye we have this part of our vision which is responsible for sharp central vision so we need to have something like this in our Thinking Machine.

Peripheral Vision

We also have cones in the side of our eye which check for unexpected sidewards motion while we foveate. So this is an early warning system for a predator and the like.

sc.vision.peripheral.vision.movement.to.the.side.check

Tpyically this is a high priority safety check. If the robot is in the reactor and doing something, then this can trigger another high priority game like duck!

Hand Eye Co-ordination

So now that we know what to foviate is, we can build up a hand-eye example. We are back to the robot in the reactor. It will have a task to grab hold of something and to turn it but first it must seek it out and hold onto it. A programmer for example can build what the robot is looking at but one needs to focus in on something specific within the frame. In this design, we break up the items into VFS items which can be reached for.

So we have a notional game GrabHoldOfSomeThing which has some VFS filter. You need to keep your vision on the item in particular during the operation.

```
game GrabHoldOfSomeThing
filter for vfs object, vfs is turnable
sc.vision.foviate.on.item.and.continue.during.operations
hos.reach.towards.item.with.hand
hos.rotate.hand
hos.hand.release.and.retract
sc.vision.foviate.on.done
```

The Thin Slice and the Virtual and Physical Self

When we are awake and walking or doing some kind of physical tasks, our virtual self and our physical self are syncd up so that our VFS is a close approximation of what is happening in the real world. A good example of this is hand-eye co-ordination where the VFS works with physical motor messaging. However our concept of our-self is a thin slice of what is going on in the body overall. There are many messages that our virtual and physical selves are not aware of. For example when we sleep there are many tasks our bodies does such as organizing our memories from the previous day.

When we rest or think, our physical self does not have to be totally synced with our physical self and so we can switch into another VFS room and do some thinking. Different games and goals needs different settings.

We can define the virtual self as a wrapper which contains our SC.

We can define the self as an Operator which contains the virtual and the physical self and contains an SC.

core.self.virtual.self.switch.to.thinking.room

core.self.name.a.b

core.self.locale.english.speaker

core.self.connect.virtual.to.physical.start.to.move

core.self.feel.pain

core.self.feel.pleasure

It's the job of the self to generate the feelings of pleasure and pain. So when we eat something that tastes good like sugar, it is the job of the self to produce this feeling. Therefore this is the core of what we think of, for example, as being human.

If humans all share the same self.core mappings we will all experience things in the same way. So we will all experience the sky as the colour blue and we'll be able to use a paint and call it blue. This is in opposition to the idea of solipsism where we only experience things. If we all share the same general DNA for the self, we will effectively have a lot in common for the most part. Clearly, not all people are exactly the same, we say some people are colour blind which means the self experiences certain colours differently and so on.

In the movie The Matrix the question is asked, how do we know what certain food tastes like? It's the job of the self to create the feeling of taste. How does the brain do this? This is a topic for another chapter but the point is that the self does this using pain and pleasure mechanisms which have been identified using things like dopamine and so on which affect brain chemistry messaging rates and so on.

An important game which the self runs is identifySpecies and this game is required for sexual reproduction and socalization. So it's important that the Self knows what is like itself and what is not.

self.identify.individual.check.is.same

This is the core to herd instinct where animals know other animals like them and work as groups in certain Interaction Games like evading prey.

Defining Thinking

We can therefore define Thinking in relation to Goals and Game.

Thinking is when the Self processes Games and Goals

The Game of Hide And Seek

When we think of machine intelligence we think of certain games that we would like a computer to run and figure out without telling it what to do. A favourite game for kids is hide and seek and this is actually an AI game. We need to understand the area we are in and we need to hide or seek.

sc.game.hide

sc.game.seek

Once more we end up with a set of games which can satisfy the solution.

Game HideUnder

Game HideIn

Game HideBehind

Game SearchUnder

Game SearchIn

Game SearchBehind

Each of these games has a payout function which is turned into an effective currency.

Game HideUnder may return a total cost of 100 because the location is too far away.

Game HideBehind may return a cost of 1000000 because there is nowhere to hide behind in view

Game HideIn may return a cost of 25 because there is a room with something to hide in

Similarly, we might have a cost for a Search and we may search locally because it takes the least effort and the cost is the smallest.

Now, in order to search in, under and behind we need the VFS to support these operations.

vfs.room.search.behind.then.search.under.then.search.in.until.found.else.give.up

To be able to search behind a building we need to have a virtual self with a position and to know which objects are close and to figure out which ones have the lowest cost. So when we look at a scene we need to build a map of the room with objects and understand their spacial relationships and how far we are from them and so on. Many video game engines already do this but what this Thinking Machine design offers is a way to tender the choices and pick the one with the lowest cost to the system and to encode the logic of this in a Visual Functional Script. Also, a new Game can be added easily.

The traditional way one might do this is to hard code some logic that runs each time with some if/then/else clauses where there are built in assumptions. This will work but will not really be an intelligent design.

The Game of Peek-A-Boo And Learning New Games

A key skill and AI learning game that children with normal development is to play games like Peek-A-Boo. The idea here is to teach a game by direction and show the rules of the game. In children with Autism for example, they do not look at a teacher or adult in some cases and will not respond to their name. In this case we can see the importance of Interaction Games where ones teaches a new game. It does not have to be Peek-A-Book but it could be learning how to play soccer or any sport by watching.

Interaction Game LearnByWatching

vls.record.then.categorize.then.visualize.and.replay

This way a child or even an adult can learn new games which we think of as behaviors or skills. It's important that games are categorized, for example, one may learn new search techniques and then this is an optional game which one can use if required.

Game SearchBySquares

Category missingPersonSearch

team.interaction.game.break.up.search.space.into.blocks

This might be a good game to use for a certain type of incident and not necessarily for hide and seek in a house.

Dynamic Contexts And Solution Paths

Our typical concept of AI is based on our ability to learn some kind of map which includes some kind of obstacles and where there is one or more solution sets. In a perfect world, all we have to do is do some kind of Interaction Game where we read or watch a show and learn how to solve a problem. Consider the case of a mouse dropped into a pool of water where there is a platform where it does not need to swim. With some coaching, or by searching the mouse finds the platform and rests there. When it is dropped in again it looks out for the platform.

sc.learn.map.room.pool.trainer.task

sc.build.context.water.tempature.x.and.container.y.dropped.into.and.swim

sc.search.and.find.platform.and.rest.then.picked.up.by.trainer or

sc.interaction.game.trainers.shows.platform.then.rests.and.picked.up.by.trainer

We can build a game from this

sc.save.in.game.context.water.temperature.x.and.container.y.where.solution.path.is.swim.to.container

Game AtoB

context

water.temperature.x.and.container.y.dropped.into.and.swim

solution

sc.find.and.swim.to.platform

The map is composed of VFS Objects which have certain behaviors.

VFS Objects And Behaviors

So we have objects that have properties. Some may contain another SC but not all do. Some we can just move or pick up or act upon another VFS object in some way. We learn these behaviours of these objects over time.

VFS Object Chair

can.move.from.a.to.b

can.knock.over

can.lift

VSF Object Door

can.open.by.turning.handle

can.close

has.dimensions.x.y

leads.into.room.c

So we can use Functional Scripting to explain what it does in an Action Script kind of way. So if we are a robot in a reactor and we see a door handle. If we have a smart enough VFS object recognizer/ generator which can identify this object, then we know the type of games we can apply to it.

```
Game AtoB
context
door.hande
can
open.door
close.door
etc;
```

We can tender for the best Game to solve our requirement such as Enter Room. Typically the game we will pick is the Game which knows how to Open a Door by twisting a handle or pressing the handle down! However, maybe the door is jammed so we have to try another game with the next best pay-off like climb in a window and so on.

Finding Related Games

A valid question one can ask is do we have fixed links to the possible games or can we grow connections dynamically. This goes back to the discussion about Static Games versus Dynamic Games. Put simply, we are able to "grow our own games" dynamically. If we only have static games that is more reminiscent of hard coded software programs which might suit certain routine applications and might be how certain illnesses damages brains in certain ways where behaviour is very static. The person can function but the range of behaviours is limited. Of course if we can figure out which Goals and Games are damaged or missing and where, there is a chance we can fix this. Dynamic Games naturally put us at the top of the food chain so if we need to give a machine dynamic games they should be in equilibrium with us like to Shut Down The Nuclear Reactor and Save The City! Then the Thinking Machine can be our hero and help us!

Personal Space And The Force Field

People have their own psychological concept of a personal space. The definition of personal space is; the physical space immediately surrounding someone, into which encroachment can feel threatening or uncomfortable.

This is an early warning system. If someone is too close then one can feel under threat. Any interaction in this type of game has low levels of trust and is a high priority message to the SC.

self.sc.personal.space.invade.monitor.any.physical.contact.with.high.priority

Pick pockets do this and might tap you on the shoulder and so on while your pocket is picked for example.

There are all kinds of rules about how close people should stand from one another and so on. There are rules for air-planes, sub-way stations, bars, how to sit in a cinema and so on.

Interaction Game

Cinema

Find empty seat but not right beside someone

Interaction Game

Plane

Talk

Do not look stranger straight in eyes while talking

Making visual eye contact with other people can also be awkward for social situations when one does not know the other. This is covered under established Social Rules.

Placing VFS Objects In A Room with a Map

A simple example is starting work in a new company. The person brings you around the company, shows you where the coffee station is and then your desk. Then the work you will do will be explained to you.

sc.interact.with.co-worker.and.learn.new.map.office

sc.vfs.position.chair.left.corner.of.room.x.y.z

sc.vfs.position.office.center.of.office.x.y.z

sc.vfs.office.lighting.record

sc.vfs.watch.manager.do.strange.dance.at.party

and so on. All of this information is stored in visual memory and call be recalled later.

Chapter 28 Aiming Not Just For Good Design But Great Design

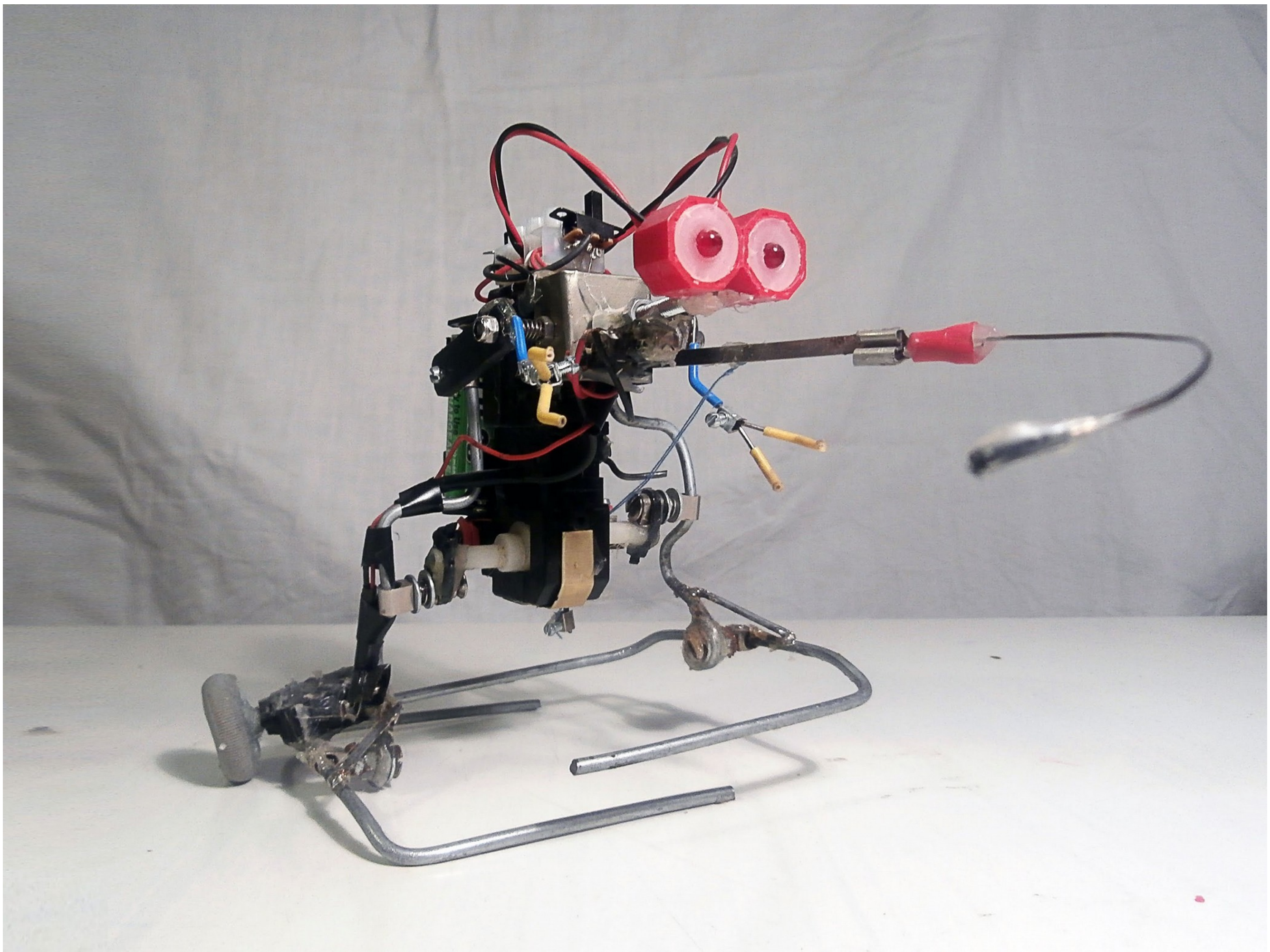
Introduction

In this chapter I will discuss a more esoteric topic but an important one and it's to do with Great Design not just Good Design. I'm talking about design that **inspires.** When people try to build machines invariably they look pretty awful and are uninspiring. I guess, if it works it works! Right?! Well if one looks at Fortune 500 Companies the image is also as important as the fact that it works and that's why they are where they are. Let's talk about this rather esoteric topic that's important in my opinion.

It's not just about the technology

In the 1980s there was a consumer boom in personal computers. Robotics is at this stage too and maybe a little before it but it's growing yearly. Mostly on a home budget, you can wire up a computer to a breadboard and get some electronics to do some kind of project. Drones of all kinds are on the rise and the price gets cheaper each year. Companies are buying machines which can run like animals. Right now, we don't have true AI on the home-scale but mobile phones are also introducing early AI personalities on their phones, seeking to assist their users. Over time one can imagine that mobile device personalities can enter a Thinking Machine phase. In time they will be kinda like our PA friends. The robots look pretty awful but we're in the early stages. So the question is, what are we looking for in a robot or a thinking machine. Let's take two examples. Theoretically both work.

In the first case, we have a home build of a robot which walks and is kinda cute.



Or we could aspire to this. This is one imagined in Isaac Asimov's Robots and Empire.

Although this is cooler, it looks like us and could spook people and fear for their employment. I'm not saying what the right choice is but it's a vision thing. In the 1980s there was a battle over the hearts and minds of people for computers between Apple and the PC and market-share. It's an old story now but it's pretty clear that Apple has won the hearts and minds of consumers although Windows PC are still moderately popular but it's hard to find anyone who gets really excited about a Windows machine like they do about Apple hardware.

So if we're going to build a Thinking Machine do the aesthetics matter? In the short run, no but I think in the long run yes. Imagine what a robot would be like if Steve Jobs had created one in his lifetime. This is a thought experiment based on how he launched his Apple Mac back in 1984. He unpacked the clunky computer and it spoke to the audience. People clapped and cheered. So using a Steve Jobs analogy, if he launched it, it would probably introduce itself to the audience and detail what it could do and chat with Steve. You'd have Steve Jobs on stage with an Android doing some cool stuff and the Android would look awesome maybe like an Isaac Asimov android. In 1984 the crowd went mad for the Macintosh.

Now imagine another company launching their android explaining their competitive price point and how it's better value and can also do as much and listing out all the features one at a time. A cool video might play and someone who had rehearsed their lines would talk. The Android might look a little clunky maybe like a budget car and the crowd might be a little interested.

Clearly we have the 'Steve Jobs effect' who as a personality is hard to compete against. He was revered and still is. It's hard to replace that combination of vision and salesmanship. Steve was a born salesman but he loved design. He once said, it's not about the technology, it's about what the consumer wants. In truth it's all about all of the above but he placed the consumer at the heart of the discussion on stage and engaged them but the consumer assumed it would be good because of Jobs. However, in truth you need good tech even if the robot is shiny and cool so the idea is that this doc has some good pointers in the right direction for robotics and AI.

So far, I've outlined what I think of as the cool under-the-hood technology although many of the specs are a little vague but if we're looking for something really cool to work on then the look and feel of the Thinking Machine is very important in terms of maintenance and reliability. People have limited budgets but we're aiming for something easy to crack open and fix.

So what we are aiming for is a Thinking Machine which extends and complements people's values rather than competes against them but it's clear there will be strong resistance to AI and with good reason but it's clear if humanity want to make it to the next stage in its evolution, we're going to need Thinking Machines to explore the Solar System and beyond for us due to current limitations on our engineering and Physics.

So let's just say we go for it as a species for Thinking Machines and go for the high ground in terms of design. Some examples are: It could be anything from a nurse requiring assistance with a patient to a tech who has to maintain a fighter jet and finds the design under the shell of the jet as nice as how it looks on the outside. The design makes sense and does not crash a lot and all the bits are easy to get at and replace. To a certain extent it goes back to the idea of Equilibrium between Humans and Machines. Ideally we're looking for people to get excited about Thinking Machines the way they get excited about awesome cars, jets and magnificent mobile phones and computers.

Ideally if you would would work for a company developing AI and robotics and you ask them what they do, they might say We're Making Life Better For Everyone or We're Helping Our Species Evolve instead of saying We're Building Machines To Take Everyone's Jobs. It all about how we are going to place Thinking Machines in our society whether they will excite us or not. It they don't I could imagine human society banning them like unwanted weapons' technology.

The way I see it for this to work is, we need Thinking Machines to be like our best friends. Maybe this sounds cheesy and naive but we need the aspects of a good friend in a thinking machine form for this to work. This stuff is as important as using Visual Functional Scripting and Category Theory in my opinion.

Overall Architecture of Intelligent System

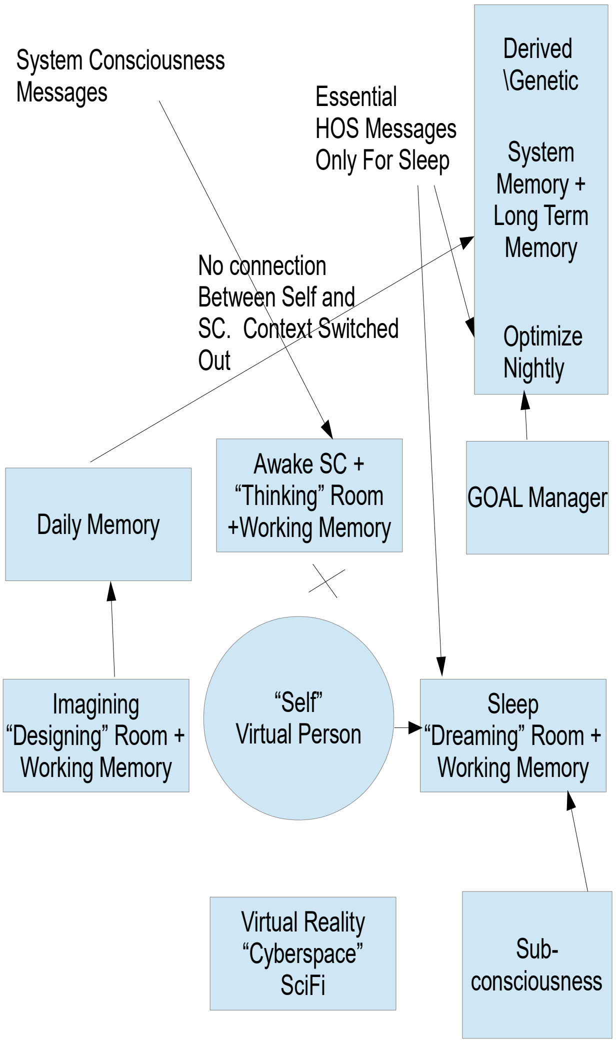
Now that I have covered all the major topics, I will describe the overall architecture of an intelligent system. This does not have to be an Android. It can be any kind of device that could be made intelligent. The design pattern roughly matches how our type of intelligent life works. Obviously it is not a perfect copy. Our life and the life around us took millions of years to evolve, adapt and self-improve so this is a super-simple version of how we work according to this theory.

Core Concepts

1. When we are asleep our bodies not only heal but optimize our memories and GOALs.
2. Our brain has the ability to create rooms which match reality and have working memory where logic and reason is performed in relation to this room
3. Our view of reality is called the SC RR Room (Reality Room)
4. There is a Sleep Room for when we sleep and there are also multiple Creative Rooms for imagination. Each room has its own Working Memory.
5. Our working memory is copied into our Daily Memory which has only room for a couple of days before it needs to be cleaned out
6. Our Long Term memory is filled with Daily Memories when we are asleep during REM and Daily Memories are cleared outcomes
7. Our Long Term memories are optimized by Neurotransmitters in Deep Sleep
8. Our Goal Manager maps our Dopamine Neurons to our Long Term Memory and optimizes them when we are asleep
9. Our SC (System Consciousness) pairs certain Message Types e.g. Audio and Sound
10. Our SC RR room has a Virtual Copy/Self of the body of the system in question and both are synced with each other
11. The Virtual system and the Physical System are placed on two separate circuits for efficiency reasons. Therefore it is possible to to plan virtual moves without having to do Physical moves permitting mental Planning.
12. Actionable Message Types are paired with related Apps to perform the logic related to the Message in question.

13. Our GOAL manager filters for pattern matches in our SC. If a close copy is found to a GOAL then the percentage of success is calculated by a GOAL App and the appropriate pathways (physics and virtual) are activate to try to achieve the GOAL.
14. In a system with two separate brain cores/hemispheres for each side of the being there is a need to handle synchronization which covers human rhythm between cores such as locomotion on two limbs if this is the chosen physical form.
15. Learning is a mapping function to the locale for speech, reading, writing and so on. The beings must have a Core OSL which is independent of the Locale Mappings. Learning mode is initiated by Actionable Messaging.
16. Logic and Reason is computed in the appropriate Room where the Virtual Self is located and is based on Traditional Set Theory applied to appropriate Message Stories.
17. A Room supports both 2D and 3D OSL Messages for the Virtual Person.
18. Each object that uses this design pattern shall have an associated Virtual Person that is Socially Aware and understands its Social Status and Grouping depending on its function.

ASLEEP!



System Consciousness Messages

Essential HOS Messages Only For Sleep

Derived \Genetic

System Memory + Long Term Memory

Optimize Nightly

No connection Between Self and SC. Context Switched Out

Daily Memory

Awake SC + “Thinking” Room +Working Memory

GOAL Manager

Imagining “Designing” Room + Working Memory

“Self” Virtual Person

Sleep “Dreaming” Room + Working Memory

Virtual Reality “Cyberspace” SciFi

Sub-consciousness

Appendix A

Software Prototypes

At present the location of the software prototypes is

https://github.com/thinkingmachine/thinkingmachine

The prototypes covers examples of the Replicate Refine patterns and the cost and payoff algorithm.

There is also some developing work using Akka to model the Replicate Refine pattern and message passing using Games and Goals.

Software Language To Achieve Design

Functional Language: Presently, the preferred solution is Scala with Actors/AKKA (or something similar)

The language chosen needs to be a functional one. By this I mean, one needs to be able to model a massively parallel solution so the code should focus on each job that the system needs to do and have it co-operate via messaging. One can thinks of a nervous system as a message bus. Each cell is a Scala Actor which can accept and send messages. The messages themselves can be thought of as the State for the Actors. If you are unsure of what a Scala Actor just lookup Erlang or AKKA.

The idea is NOT to write the whole program just create the Actors who are time dependent and handle message passing. From a coding perspective we need to focus on the job of each actor (similar to a specialize cell) and what are its messaging options are based on the system time and the actor creation time. This is analogous to cell birth inside a body, function and `death. Each actor is aware of its neighboring actors and knows what should be next to it. So it we get damaged at this point, pain messages are sent a message is sent to rebuild new neighbors. Therefore one can think of cancer as a messaging problem possibly to do with with incorrect timers or general faulty messaging where a cell never dies, for example.

So we do think in terms of the sequence of operations required to operate the Android we model a series of co-operating Cells/Actors exchanging messages. There is of course a hierarchy for the cells. Each Actor knows its cell type.

Importantly, each Cell/Actor contains a Clock and it is important that all the Actors have the same time. The reason for this is that the responses of the Actors is also time dependent. So each cell knows for example when it is first created how old it is.

Therefore each actor knows

What the overall system time is (This would be like a date of birth/conception)

How old this actor cell is

Therefore, if it's just “new born” there are some setup tasks.

Once it's setup, it accepts certain messages and then sends out other ones

Each cell knows the cells that are beside it so there is Geo-locational knowledge

For example, if a neighboring cell says “I am damaged” or “Stops sending a message” then it can request a new cell is created in this location.

Vision processing should also work like this where each Actor processes a small part of the overall image and passes messages onto the next actors.

For example, the actors might filter a certain color and then send a message to the adjoining actors “Color found” or “Color NOT found”

Ultimately this message travels to the visual processing system which builds a three dimensional view of the messages.

It also important that for example a virtual system can operate without every message and work with what it has. For example, if one is deaf, then one can still see and the body continues to function. Using this design pattern we have a very redundant design and one which can take advantage of a multi-core processor solution in an android and not be concerned about concurrent issues and or locking. Not all messages/organs need to be present to function.

Software That Builds Itself

In this design pattern, a single Actor Cell can build the entire Cell program. To do this requires the idea of a program that grows what it needs over time analogous to a startup phase.

A simple example is a memory storage program.

We can start with the starting Cell and it builds as many storage Cells that are needed. The number of Memory Cells grows over time.

Therefore the actual memory of a computer is used to support the Cells but the data is not stored directly in them by the Cell rather abstractly. Instead it is stored in the Memory Cells. The CPU and Memory of the machine are like our reality where we can create new cells inside it beside one another and coupled together in some way.

We need to develop software that is aware of its own stages of development.

CELL DEVELOPMENT ALGEBRA

Start

Grow

Mature

Reproduce

Continue

Complete

To do this each Cell needs to keep track of some kind of timer t and have some kind of DEVELOPMENT ALGEBRA forming the life-cycle of the system.

CELL STATE ALGEBRA

Cell Wait in State

Cell Move to next State

These waits might be represented in Milliseconds

Appendix B

Hardware To Achieve Design

Current thinking is prototyping on something like a Raspberry Pi but does not have to be – depending on resources and budget. The messaging software comes first.

Appendix C

Message Algebra Types Requirements And Related App(s)

An App supports specific messaging which is how the App Cells communicate with one another. An App is responsible for building and creating the physical structures of the App. It is responsible for the total life-cycle of the App over the many cells. For example, the Reproduction App builds an entire being from a starting cell. The App knows what it is building. One App may call another App during its life cycle. A Cell may support one or more App messaging protocols in the Cell body. Each type of Message Type has a unique identifier.

e.g.

Inter cell messaging (One Neuron to another)

Visual.Algebra.<FromCell>.<Message>

Visual.Algebra.<ToCell>.<Message>

Intra cell messaging (Core to Runtime)

<FromApp>.<Message>

<ToApp>.<Message>

Type	**Requirement**	**App (DNA Application)**
Visual Algebra (Ventral)	Vision Related	Facial Recognition Stereoscopic Vision
	Recognition:	Pattern Recognition

	Face Image Color Letters Shapes Objects Movement Depth Back Light setting Shadows 2D Messaging “Visual Cartoons” 3D Messaging Stereoscopic vision Face Nose Face Ears Face Eyes Face Cheek Face Chin Face Lips Face Brow	Color Recognition Shape Recognition

	Face Hair Focus Algebra: Focus on object (in Room) Track Object Jump to Object (Peripheral Vision)	
Audio Algebra	Audio Related Recognition: Word sounds Music General Rhythm Pitch Tempo Vocal Chords	Audio
Language Algebra	Verbal Communication Singing Speaking Sounds	Language
Action Algebra	Virtual Movement	Self

	Physical Movement	
Sensory Algebra	Central Nervous System and Peripheral System Hand Skin Touch Organs	Sensory
Emotional Algebra	Limbic System Fear Flight Fight Freeze	Emotions
Olfactory Algebra	Taste, Smell and Airborne Messaging Happiness Sadness Disgust Salty Sweet Sour Bitter	Olfactory

Improvement Algebra	Evolutionary Improvement Messaging Pain Improvement Suggestion	Improvement
Development Algebra	Reproduction, Conception, birth and growth Sex Trimesters Maternity Organ developmental Bone development Ageing Skin Temperature Core Temperature Skin Pressure Core Pressure	Reproduction
Expression Algebra	Facial Expression and Body Language Smile	Expression

	Sad Laughter Head shaking	
Autonomic Algebra	Autonomic System Heart-beat Blood pressure Temperature regulation	Autonomic
Logic Algebra	Logic, Reason and Math Semantic reasoning	Logic
Core Algebra	Mechanism and rules by which messages combine AND OR NOT SQL Syntax	Algebra
Consciousness Algebra	Conscious thinking Virtual Self Placement Goal actions	Consciousness
Goal Algebra	Goals of a being Dynamic goals	Goal

	Static Goals	
Cellular Algebra	Cell Messaging Master Program Cell State Cell Age Inner cell messaging Outer cell messaging Cell type	Cell
System Conscious Room Algebra	Defining 3D rooms and switching between them Create Switch Room Messaging	Room
Motor Algebra (Dorsal)	Muscle read and write Read muscle Write muscle Proprioception Move Arm Move Leg Move Hand	Motor

	Move Finger n Move Thumb Move Head Move Shoulder etc; Eye Move x (cardinal directions) Eye Move y Eye Focus Eye Retina Eye Peripheral Eye Light Adjust Eye Blink Eye Smooth Track Eye Partial Track	
Social Networking Algebra	Troll Detection Manners Liking Disliking Public Friends Private Friends Groupings Sharing Social Interaction Protocols	Social Networking

Remoting Algebra	External communication Access Protocols For Other Android Access Protocols For Other	Remoting
Object Builder	Messaging that builds a physical 3D structure from a Virtual 3D Object Builder Messaging	Builder
Master Planner	Overall blue print for a body Orchestra Messaging	Master Planner
Sensor Algebra	PAIN VIBRATION PROPRIOCEPTION TEMPERATURE SKIN STRETCH SKIN PRESS FINE TOUCH	Sensor
Body Position Algebra	Arms Face Legs Knee etc;	Sensor Body Position
Body Location Algebra	Arms Face Legs Heart Knee etc;	Sensor Body Location

Memory Algebra	Store chronological Backup chronological (aka REM sleep) Look up long term Store long term Optimize connections long term Switch out consciousness Switch back in consciousness Learn task procedural Apply Operation Count Intersection Union	Memory App
Math Algebra	Count Move Rotate Translate	Math App
Social Interaction Algebra	Meet Greeting Eating Business Interaction	Social App

Appendix D

Message Circuits

The general architecture of the brain involves various message circuits. This is how the Android brain will work. The message circuits will loop out from the Thalamus and then go to the areas of specialization (App Zones) and then loop back to the Thalamus. Nerves carry the message information. For example, Visual Algebra is carried on the Optic Fiber from the Eye to the LGN. Once they arrive in the SC, the messages will be processed and there may be result messages which are fed back into the circuit. A message circuit can have one or more attached App Zones. An example of an App Zone is Word Recognition which is connected to Sentence Forming and then back to the SC which is in the Thalamus.

These are simplified example of how an Android can perform a task. The idea is to create the message circuits that achieve a particular task at a high level.

The constituent parts are:

Zone Connectors (Nerves, Fibers etc;)

App Zones (Built by DNA Apps during their life-cycle)

A (loop) Zone Connector connects the **first** Zone to the **last** one.

Specialize DNA Message Algebra between Zones

The Central Hub for all circuits is the SC in the Thalamus.

Circuit Name	Related Circuit Apps (Nerves carry the information from one App Zone to another)
Vision Control	Eye SC (Thalamus LGN)

	Pattern Recognition (V1) SC (Thalamus) Eye Control Eye Focus (loop)
Motor Movement Rehearsal	Recorded Movement Putamen SC Thalamus Motor Cortex (loop)
Music Circuit	(Hippocampus) Lookup song fragment (Motor Control) Singing SC Thalamus Audio Listening SC Thalamus Tune/Adjust voice (loop)
Memory Circuit	Thalamus Working Memory Daily Memory Hippocampus PreFrontalCortex

Appendix E

Theoretical Chronology of Senses And Paired Messaging

External Sense	**Description**
Touch	External skin touch TOUCH ALGEBRA
Smell	Detecting external scent messages OLFACTORY ALGEBRA
Taste	Sampling air/water for scent messages TASTE ALGEBRA
Balance	Moving and not falling over BALANCE ALGEBRA
Hearing	Sounds Words Singing AUDITORY ALGEBRA
Speaking	Sounds Words Singing VERBAL ALGEBRA

Vision	Day vision Night vision VISUAL ALGEBRA
Sign language	Visual signs with meanings SIGN ALGEBRA
Internal Sense	**Description**
Central nervous system	Fast internal, direct NERVOUS SYSTEM ALGEBRA
Hormone messaging	Slow, internal general HORMONAL ALGEBRA

Paired Message Types

Message Requirement	**Message Types**
Vision with sound at SC	Visual Algebra Audio Algebra
Movement with sound at SC	Motor Algebra Visual Algebra
Sense with Location at SC	Sensor Algebra Body Location Algebra

Physical Self and Virtual Self at SC	Physical Message Algebra Virtual Message Algebra
Vision with Emotion	Visual Algebra → Semantic Algebra Emotional Algebra
Pain with Emotion	Pain Algebra → Semantic Algebra Emotion Algebra
Smells with Emotion	Olfactory Algebra → Semantic Algebra Emotional Algebra
Message with Date (Rare)	Bibliographic Look up of Memory

Appendix F

Software Prototypes

It's not possible to do the whole piece in one iteration so here I'll define the various tasks. All major Android pieces inter-communicate via a general Message Algebra. Each specialized region is responsible for some type of input and the output is messaging. Using an architecture like this, vision on the Android could fail or not be present but hearing messaging would still work like a person and have some kind of function unless it is a critical sub-system which fails like CPU(s).

Algebra = Messaging which can be operated on by AND, OR, NOT.

Vision Messaging	Take any photo and turn each pixel into a message queue. Generate Visual Algebra based on the pixel Message Queue patterns.
Auditory Messaging	Take any sound and turn each wave into an Auditory Algebra. Generate Auditory Algebra which can find certain basic patterns. Pitch Tempo Rhythm Key
Room Messaging	Create a Visual Room Message protocol where rooms can be created and Message Queues attached which generate images
SC	Build a system conscious process which

	handles multiple message queues and pairs some depending on configuration. Allow switching between Visual Room depending on state e.g. Day vs Night.
DNA App Store	Create an Android cell (Software Based) which has a Core that contains an App Store where certain Apps can be switched on or off in the Outer Core. The Android cell knows which ones are switched on in the run-time layer (Outer Core). The Outer part of the cell is where the active Apps run.
Cell App Architecture	Develop a base Cell Architecture for all Cells where Messages comes into and out of the Cell. This is how the Cell has internal and external communication. The messages handle Creation Replication Communication Internal/External Operation Shut down Inner Core Messaging Outer Core Messaging
Sample Apps	1. Develop a “Hello World” sample app that takes in a message “Hello App” and replies with message “Hello”. 2. Develop a sample App for the Cell App

	store which does a basic function such as detect a certain vision pattern knowing the value of other Cells. (Other Cell types with their own Apps may be required.) The App has its own Messaging which describes how to achieve this.
Test Message Pairings	1. Test pairing sound with vision 2. Test pairing sound with movement
Dual Hemisphere	Create a dual hemisphere example where each hemisphere handles certain Message tasks and some of them are synchronized across hemispheres.
Robot attachment	Create a Motor Message Story Using a Motor App for the attachment in question send the Motor messages and perform some kind of basic operation
Message Stories and Memory	Store one or more Message Stories in the Storage App and retrieve from a variety of similar messages
App Tester	Demonstrate how a Cell with a particular App switched on not only handles the messaging for that app but also builds the structure for those message e.g. A message App builds its own storage cells before populating them with data.
Future PPR Room Tester	Develop an app which will take a SC-RR and add imagined future Message Stories to determine steps and outcome in a SC-PPR

	“Imagination Mode” Simple example, goal is wash dishes More complex example, goal is board game figure out next move The SC-PPR can be based on the SC-RR or another imagined room in another advanced case.
Dance Program	Develop an app which pairs Audio Messaging with Movement Messaging. Run a test program that matches a certain type of Rhythm message to a Movement message to demonstrate a simple form of dancing.
Personality Type App	Develop an app which takes two conflicting personality types and gives them each a percentage. Run the app and show how it generates a percentage of one type of messages types versus another over time t.
Body Locator App	Send a message along a nerve pathway and identify the part of the body it belongs to
Body Position App	Send a message along a nerve pathway and determine the position of a limb or muscle e.g. lift leg up
Virtual Self App	Send a message to move a limb. Update the Virtual Self to reflect this physical position. Get a message that a limb has moved (either intentionally or unintentionally). Update the Virtual Self to reflect this physical position.

	Android (Advanced) : Push an Android over and have the virtual self generate the limb messages to stand back up again.
Physical and Virtual App	Develop a simple app which takes a simple object like a car door. Create a virtual version of it and attach sensors to the real version of it. Open the door and generate a OPEN DOOR message. Send this to the SC and route this to the virtual self of the car door where the computer generated image opens the door of the car. Advanced: Close the Virtual door of the car and have this send a CLOSE DOOR message to the SC. This is then synched with the Physical version and the door of the car physically closes. Advanced: Send a complete set of BODY POSITION messages from the Virtual Self to the Physical self and ensure that both are synched (and vice versa). Advanced: Send a MESSAGE STORY and have the physical self do some kind of operation. Note: This requires that the Virtual Self and the Physical Self messaging are paired.
Temperature App	Attach temperature sensors to an object with

	a specified range. Send stream of temperature messages to SC. If limit is reached then adopt a strategy e.g. overheat, turn on fan until back within range again. Do the same for pressure example.
Core SC App	Develop a Core SC app where one can plug-in message types and Apps via configuration.
Message App	Create a SC Create Working Memory Store Some Messages In Working Memory Copy episodic memory to daily memory Sleep Copy daily memory to long term storage In SC fetch messages from long term storage Place in Working Memory In SC take message stories and add to working memory
Day Night Messaging	Detect night Sleep Complete Sleep Tasks

	Wake Up
Message Operations	Take some Message Stories and apply the following operations Union Intersection Count Subtraction etc;
Language Translation Logic	Convert a question into Message Story algebra equivalent Create some sample questions Can a Penguin Fly (subtraction) How many cars in the driveway (count) The cat is good at maths (true/false)
Neuron Examples	Create Neuron Actor Cells Each has input queues and output queues The queue values are stored in the Cell and form the message story Take a value, store it and pass it onto another

	output queue Take cell with a value and copy it to two output queues Take several queues and store in cell then output on each output queue Take different message types e.g. visual and audio and combine them to form a combined message story
Math App	Count the number of items in a message story Add/Subtract multiple entries Visual Algebra Translate Virtual Location Rotate Object[s] Scale etc;
Optimizer App	Create a GOAL for EAT Add associated messages Filter on these messages in SC If a match is found notify Dopamine EAT neuron with percentage Check that related messages are found in SC

	Confirm goal achieved message is Reached Store in Goal Manager
Math Algebra App	Integrate a Computer Algebra system like Maxima and support Message Operations like solve and so on e.g. SOLVE FOR QUADRATIC
Message Chain	Define a Message chain and related Apps that perform a certain function.
Message-App-Validate	Invoke an App via a Promoter Message and when the App is complete check the Validation Messages for success or otherwise.
Threat Detection	Send messages to an App that detects threats like a Caterpillar is eating leaves and generate defensive strategy messages
Run time system information	Create a run time mechanism mirroring the Cytosol that keeps track of Apps running and state information in the Typed Actor Cell
App Manager	Create an App manager similar to a chromosome which allows Msg-Apps to be added to it. Create several App Managers in a Typed Actor Nucleus
Security Manager	Create a Security Manager that checks the keys of the Msg-App against its keys and from this determine if it is allowed to enter or not an App Manager.
Helper and Connection	Create a helper to ferry something from one

	place to another. Create a connection for one place to another.
Synesthesia	Allow two Msg-Apps to handle the same message and combine the results in the same App Manager
Evolution	Create an evolution style app which creates a single Msg-App and rearranges the code pieces into blocks of code. Arrange them, randomize some parts and get it to spawn new ones. Randomize Key Randomize Container Randomize Msg-App Randomize Engine Randomize Message Transcription Factors
Body Builder	Create an app that knows the physical and virtual form it is building. Break the body into core components which can be connected together in a lego fashion. Define the shape and the components Define the virtual to physical mappings Have specialized apps dealing with different pieces e.g. the eyes, the torso, the legs and so on.

	Permit flexible skinning of the body parts. [Work in conjunction with Promoter App to know what the related Cell types are.]
Hereditary App	Implement builder app that is based on hereditary rules
Transcription Factor App	Define the rules under which an App will be called. There may be multiple Messages required in order for an App to be transcribed/executed.
Engine App	Define an App which will run a transcribed App based on certain transcription factors/parameters.
Sensors App	Define an App which can determine the weight and the Gravitational constant and alter some value like the amount of, say calcium (or anything) in the cell type.
Promoter App	Define a Promoter App which contains some kind of structure which determines a tree like structure of promotional Cell Types and what the rules are to move from one cell type to another. Typically this will involve some kind of transcription messaging. If it makes sense, place the Cell Type in the Run time of the Cell unless another mechanism is preferred. e.g. stem cell becomes skin cell becomes...
Operators	Create a suite of LOS Operators that can operate on Message Sets that perform much

	of Set Theory. These can be thought of as Operator Engines. Construct the Message Sets from distinct entities which can be like molecules.
Operators related to Gaming and Goals	Create a trust operator between operator A and operator B which has percentage of trust Create a set of Bayesian operators with percentages linking them in an acyclic fashion Create a personality profile as Bayesian percentages Assign a personality profile to an operator, preferably a player Create an operator with some message set information Create a flywheel operator which can loop and has a set of operations Use a flywheel operator to consume a set of messages and work out probabilities Create a player with self-directed goal create an object with no self-directed goals

	Determine a set of operations, use the personality profile and trust between players to determine the next move/response while looking up the Bayesian network and over-riding where appropriate e.g. prisoners dilemma
Chronology Operator and Simulation Operator	Place messages in an Operator and store them based on their time of entry. Retrieve them based on their time of entry Pass messages into a Simulation Operator and have it predict future outcomes of a game without playing the game in reality.
Attraction and Gender Operators	Create an Operator that contains gender Create an Operator that contains attraction attributes
Goal and Game Manager	Create an Operator that manages the Games and Goals of the Thinking Mind Implement install/view/update/Delete operations to manage the Goals and Games that are in place. Also implement a mechanism which can monitor the Goals and Games that are running with some security features.
Physical And Virtual Planning	Create a game with multiple message solutions Apply each solution to the Virtual PPR Operator and determine payoff Pick the one with best payoff

	Apply message solution to the Physical circuit to do the task and measure the actual payoff
Effective Currency Payoff For Games	Calculate a payoff currency for a game based on system requirements which are best for the thinking machine's e.g. time energy expended safety etc;
Games with Message Criteria	Create some games which are only activated based on certain message criteria e.g. Go from A to B game where there is an obstruction Drill hole Climb over Go around Find alternative route etc;
Create an Infinity Game	Create an Infinity game based on two static methods replicate and refine which call one another. The games have an Instruction set which can

	be altered by the Specialization game. The replication game operates on the instructions given to it and calculates the payoff for the instruction set costing the tasks versus the payout e.g. move and find food, and passes the results along with the instructions to the specialization game determining whether the game is succeeding or not. Each instruction costs a certain amount of energy The goal is to find and process energy based on the instructions. The cost of building, finding and processing the food should be <= to the energy cost of following the instructions.
Tendering Game	Create a game which has more than one sub-game which can be tendered for a solution on the virtual circuit. Create the game steps and run them on the physical circuit. Monitor the actual results against the predicted continuously and refine if necessary.
Dynamic Game	Create a dynamic game which forms based on message criteria and optimizes based on best payoff
Hardware located Games	Create a hardware game with a goal dedicated to a piece of hardware maintaining a certain state. - similar to a knee keeping

	someone up.
Stress Game	Run a goal that tenders for games but none of the games achieve the desired payout and generate stress messaging
Pain Game	Create some goals and games and have the games stop unexpectedly. Generate Pain messaging and send it to the SC kicking off the pain Game.
Pleasure Game	Monitor the amount of games which are either meeting or exceeding their payouts. If they reach a certain threshold then generate pleasure messaging which kicks of the pleasure game.
Create a board game based on two Group Interaction games.	Create pieces and moves Create referee to determine winner Create a virtual board and a view which is a physical board Later Create Player 1 and Player 2 to take turns. Give each player some kind of personality type to ensure moves are different.
Build a stand-alone device that has its own SC and is not a robot but a limb of some kind.	Create the limited Games for this limb Create its dedicated SC

Appendix G

Goals and Related Behaviors

If we create an Android without any goals, it will just stand in place and theoretically do nothing. Therefore it's important that we have predefined goals for the Android. People are no different and many of the basic goals are shared. Also, if a goal is not achieved there are certain responses and vice-versa if achieved. Maslow created a hierarchy of needs / goals which matched many of the brain's GOAL Algebra. This is a limited subset. An Android should create dynamic goals where appropriate and many should be cascading based on others.

Basic Goals	Description
Food	For an Android this is some kind of energy source.
Shelter	Some kind of home.
Safety	A place that is safe to work.
Job	A task to be done.
Friends	Maybe human and/or other machines.
Activities	Sports etc;
Family	Maybe a company or family droid.
Health	Regular maintenance.
Reproduction	More of an advanced feature when Androids evolve.
Self Esteem	Treated properly in the work place and relied upon.
Peer Recognition	Trusted friend.

Eating	Food gathering
Mating/Breeding	Next generation preparation
Finding a home	Establishing a place of residence

If a goal is not reached, the body typically reacts in some kind emotional way in an attempt to achieve that goal.

Unattained Goal	Description
Stress	
Fear	
Anger	
Flight	
Fight	
Panic	
Dispute	

If a goal is achieved the body is flooded with a hormone and a different kind of emotional response. It makes sense that an Android should feel good if a goal is achieved.

Attained Goal	Description
Happiness	

Goals need to be followed up by Message Stories to achieve these goals.

GOAL → TODO MESSAGE STORIES → SUB-GOAL...

Goals drive message stories which in turn highlight new goals.

Appendix H

Miscellaneous TODO Topics

//Lifeforms that transform themselves, mimics, dynamic messaging and apps (butterfly) – using for hunting and hiding

//Enhancing a life form shape e.g. exoskeletons

//Logic paths – logic messaging

//Autism - Explain Syncing Sound and Vision (Sound processing is too slow)...

//Frame rates of movie should match frame rates of mind

//Children that do not respond to their name called (Auditory stimulation)

//Internalized behavior due to poor messaging (name response)

//Herbs

//Immune system mechanisms

//Bipolar Disorder

//Does an Android need Blood/Hormones/Hypothalamus?

//Double vision

//Running multiple games of chess mentally at the same time. A "souped up" multi-room SC.

//Cerebal Palsey

//Fat Types (Energy / Heat)

//Blood Pressure

//Multiple Personality Disorder

//3D Vision

//How to slow down heart beat, deep diving

//Falling into cold water, oxygen deprivation

//Narcolepsy

//Turrets Syndrome

//Why medicines work better on some than others

//Irradiation illusion

//Motion sickness – body expels food thinks it is at threat?

//Coma – what parts can we check for consciousness if someone goes into coma

//Association cortex – where am I in the mind

//Present bias – get something now versus in the future

//Nucleus accumbens fear, reward, addiction – food and sex – money

//Tulip bulb fragrence – bulb fragrence financial bulb – possibly caused by olfactory messaging

//Cromagnon vs neanderthal

//Protocol droid, investigate how communication using good manners is employed by an Android.

//How Android's inter-communicate and how Users can “jack in”

//Humor

//Meditation

//Moral Algebra - Codes of permitted behaviors – Isaac Asimov Robot Laws

//Message Story 'Strategies' and '~Tactics' for different situations

//Define Semantic Algebra (Logic And Reason)

//Mech hazard protocols / safety first

//Defining learning as a mapping layer from core algebra to locale algebra e.g. Alphabet and Numbers

//Evolution and Self-Improvement messaging

e.g. How Chromosome 2 possibly contains a folded chromosome from our ancestors

How to architect Self-Improvement messaging

e.g. map Messages into the Chromosome

Dominant Genes; one gene versus another

How can apps and messaging evolve / improve

Exoskeletons – enhancing ones form in different ways

Extending ones brain to connect remotely etc;

Messaging types required etc;

Mimics for hunting, mating and hiding using lifeforms are examples; parrot, sea life, insects that pretend to be twigs etc;

Dynamically changing ones form e.g. a butterfly, a frog and how could a person do this?

//mitosis

centromeres

kinesin types/motors

//gene expression

//techniques

DNA Activation

Transgenic Organism

//Interspecies communication

//General design pattern for life – all life is connected via messaging. Some messages are beneficial others are not. Squid that contains glowing bacteria and communicates with them. So a complex organism makes use of its own messaging and that of helpful bacteria apps. The goal is reproduction of the cell type – that is the most basic goal.

//Counting messages

Quorum sensing – sensing how many there are before turning on apps. Intraspecies message language. Inter species messages. How many of me and how many of you. Doing tasks together before virulence.

//Multi App approaches – Dynamic threat assessment in plant, reason for more genes identify the precise threat e.g. a caterpillar and have appropriate responses

request reinforcements e.g. summon predator of caterpillar; enemy of my enemy is my friendly

change strategy completely e.g. from moth to hummingbird – have capacity to do this

a pallet of dynamic responses

//the ability to create a new response

- trying new things
- learning from existing attacks
- encoding success in the genome

//RNA vs DNA

DNA offer the ability to have run time information

RNA on its own does not support that

//Building and growing a body

Bones development and detecting bone density

Trabecular bone

Weight and pressure = Gravity detection

//engines

engines with folding mechanism, engines which fold 3d structures

engines which move things

engines which contain building blocks

engines which assemble ribosome

origami mechanism - molecular programmers, short strands bind long strands, using heat

ribosome making proteins

constantly scanning the dna in order to bend and start transcription

#TODO

Damaged physical self but virtual self still aware and mapped and ghost pains

Bionics e.g. how the foot works all the way up, arm etc; physio and the like, how massage heals the brain connections after injury

Connecting Messaging to external device, external limbs

How the brain maps its virtual image to its physical image, what part theoretically does it

Extending the body to have an exo-skeleton

Extending PPR to include external devices / machines for remote control

Regrowing a limb

if dna is static then there must be dna checkers that know start and end points for genes and related genes

engines which monitor the dna genes and related ones, dna managers engines

Appendix X

Health and Safety

Ideally there would be a framework where all static goals are declared and defined by anyone who designs a machine in the public domain so that it can be shown that it passes safety tests. By knowing the Goals and Games of the device in question, there can be a form of certification. It must be declared where a machine is using Dynamic Goals as this makes its behavior more human like.

Machine Rights and Dynamic Rights

Over time as this technology matures towards sentience it makes sense that a Framework of Rights is put in place to ensure a truly free machine in the way that we knows it by placing the same responsibilities that we would on any citizen to abide by the laws and provide some form of service to society.

Social Stability

Clearly it does not make sense to undermine social stability by replacing peoples' jobs so this must be factored in. Equilibrium should be sought where the benefit is clear. For example, Space Exploration and Initial Colony Setup makes sense for Intelligent Machines imo.

Appendix Y

The Choice of a Software Language

The document does not mandate for any particular programming language. Over time they may change and so may their capabilities so it doesn't make sense to recommend one over the other. However **Threading Matters!** It cannot be ignored. One may for example choose C++ over a language like Java because C++ **is faster**. However, this is a false economy if one is trying to create a truly parallel implementation which uses an Actor based model. By this I mean, one may see the same code initially run faster but one is still constrained by the Threading model of the language. The recommendation is to use a Programming language which is stateless like Scala, Erlang or libraries which support Actor based models where creating a new Thread is no big deal. If one does not do this then one will quickly find that the system begins to tie itself in knots as different process fight for the same resources and it will slow down. This is not to say that an Actor model will not suffer from this, however the point is that this design pattern does its level best to treat each actor like it has nothing to do with another actor from a Threading viewpoint and messages are the common links. You will of course be constrained by the power of the core[s] of the underlying hardware system.